Practitioner's Docket No. HAIR-22 **PATENT**

Preliminary Classification:
Proposed Class:
Subclass:

NOTE: "All applicants are requested to include a preliminary classification on newly filed patent applications. The preliminary classification, preferably class and subclass designations, should be identified in the upper right-hand corner of the letter of transmittal accompanying the application papers, for example 'Proposed Class 2, subclass 129.' " M.P.E.P. § 601, 7th ed.

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Box Patent Application
Assistant Commissioner for Patents
Washington, D.C. 20231

NEW APPLICATION TRANSMITTAL

Transmitted herewith for filing is the patent application of
Inventor(s): Arthur R. Hair, Christopher Gorski, Charles Greiner

WARNING: *37 C.F.R. § 1.41(a)(1) points out:*

"(a) A patent is applied for in the name or names of the actual inventor or inventors.

"(1) The inventorship of a nonprovisional application is that inventorship set forth in the oath or declaration as prescribed by § 1.63, except as provided for in § 1.53(d)(4) and § 1.63(d). If an oath or declaration as prescribed by § 1.63 is not filed during the pendency of a nonprovisional application, the inventorship is that inventorship set forth in the application papers filed pursuant to § 1.53(b), unless a petition under this paragraph accompanied by the fee set forth in § 1.17(i) is filed supplying or changing the name or names of the inventor or inventors."

For (title): METHOD AND SYSTEM FOR ESTABLISHING A TRUSTED AND DECENTRALIZED PEER-TO-PEER NETWORK

CERTIFICATION UNDER 37 C.F.R. § 1.10*
(Express Mail label number is mandatory.)
(Express Mail certification is optional.)

I hereby certify that this New Application Transmittal and the documents referred to as attached therein are being deposited with the United States Postal Service on this date November 10, 2000, in an envelope as "Express Mail Post Office to Addressee," mailing Label Number EL707030876US, addressed to the: Assistant Commissioner for Patents, Washington, D.C. 20231.

Tracey L. Milka
(type or print name of person mailing paper)

Tracey L. Milka
Signature of person mailing paper

WARNING: *Certificate of mailing (first class) or facsimile transmission procedures of 37 C.F.R. § 1.8 cannot be used to obtain a date of mailing or transmission for this correspondence.*

***WARNING:** Each paper or fee filed by "Express Mail" **must** have the number of the "Express Mail" mailing label placed thereon prior to mailing. 37 C.F.R. § 1.10(b).*
*"Since the filing of correspondence under § 1.10 without the Express Mail mailing label thereon is an oversight that can be avoided by the exercise of reasonable care, requests for waiver of this requirement will **not** be granted on petition." Notice of Oct. 24, 1996, 60 Fed. Reg. 56,439, at 56,442.*

1. Type of Application

This new application is for a(n)

(check one applicable item below)

☒ Original (nonprovisional)

☐ Design

☐ Plant

WARNING: ***Do not*** *use this transmittal for a completion in the U.S. of an International Application under 35 U.S.C. § 371(c)(4), unless the International Application is being filed as a divisional, continuation or continuation-in-part application.*

WARNING: ***Do not*** *use this transmittal for the filing of a provisional application.*

NOTE: If one of the following 3 items apply, then complete and attach ADDED PAGES FOR NEW APPLICATION TRANSMITTAL WHERE BENEFIT OF A PRIOR U.S. APPLICATION CLAIMED and a NOTIFICATION IN PARENT APPLICATION OF THE FILING OF THIS CONTINUATION APPLICATION.

☐ Divisional.

☐ Continuation.

☐ Continuation-in-part (C-I-P).

2. Benefit of Prior U.S. Application(s) (35 U.S.C. §§ 119(e), 120, or 121)

NOTE: A nonprovisional application may claim an invention disclosed in one or more prior filed copending nonprovisional applications or copending international applications designating the United States of America. In order for a nonprovisional application to claim the benefit of a prior filed copending nonprovisional application or copending international application designating the United States of America, each prior application must name as an inventor at least one inventor named in the later filed nonprovisional application and disclose the named inventor's invention claimed in at least one claim of the later filed nonprovisional application in the manner provided by the first paragraph of 35 U.S.C. § 112. Each prior application must also be:

(i) An international application entitled to a filing date in accordance with PCT Article 11 and designating the United States of America; or

(ii) Complete as set forth in § 1.51(b); or

(iii) Entitled to a filing date as set forth in § 1.53(b) or § 1.53(d) and include the basic filing fee set forth in § 1.16; or

(iv) Entitled to a filing date as set forth in § 1.53(b) and have paid therein the processing and retention fee set forth in § 1.21(l) within the time period set forth in § 1.53(f).

37 C.F.R. § 1.78(a)(1).

NOTE: If the new application being transmitted is a divisional, continuation or a continuation-in-part of a parent case, or where the parent case is an International Application which designated the U.S., or benefit of a prior provisional application is claimed, then check the following item and complete and attach ADDED PAGES FOR NEW APPLICATION TRANSMITTAL WHERE BENEFIT OF PRIOR U.S. APPLICATION(S) CLAIMED.

WARNING: *If an application claims the benefit of the filing date of an earlier filed application under 35 U.S.C. §§ 120, 121 or 365(c), the 20-year term of that application will be based upon the filing date of the earliest U.S. application that the application makes reference to under 35 U.S.C. §§ 120, 121 or 365(c). (35 U.S.C. § 154(a)(2) does not take into account, for the determination of the patent term, any application on which priority is claimed under 35 U.S.C. §§ 119, 365(a) or 365(b).) For a c-i-p application, applicant should review whether any claim in the patent that will issue is supported by an earlier application and, if not, the applicant should consider canceling the reference to the earlier filed application. The term of a patent is not based on a claim-by-claim approach. See Notice of April 14, 1995, 60 Fed. Reg. 20,195, at 20,205.*

WARNING: *When the last day of pendency of a provisional application falls on a Saturday, Sunday, or Federal holiday within the District of Columbia, any nonprovisional application claiming benefit of the provisional application* ***must*** *be filed prior to the Saturday, Sunday, or Federal holiday within the District of Columbia. See 37 C.F.R. § 1.78(a)(3).*

☐ The new application being transmitted claims the benefit of prior U.S. application(s). Enclosed are ADDED PAGES FOR NEW APPLICATION TRANSMITTAL WHERE BENEFIT OF PRIOR U.S. APPLICATION(S) CLAIMED.

3. Papers Enclosed

A. Required for filing date under 37 C.F.R. § 1.53(b) (Regular) or 37 C.F.R. § 1.153 (Design) Application

46 Pages of specification

1 Pages of claims

9 Sheets of drawing

WARNING: ***DO NOT*** *submit original drawings. A high quality copy of the drawings should be supplied when filing a patent application. The drawings that are submitted to the Office must be on strong, white, smooth, and non-shiny paper and meet the standards according to § 1.84. If corrections to the drawings are necessary, they should be made to the original drawing and a high-quality copy of the corrected original drawing then submitted to the Office. Only one copy is required or desired. For comments on proposed then-new 37 C.F.R. § 1.84, see Notice of March 9, 1988 (1990 O.G. 57-62).*

NOTE: "Identifying indicia, if provided, should include the application number or the title of the invention, inventor's name, docket number (if any), and the name and telephone number of a person to call if the Office is unable to match the drawings to the proper application. This information should be placed on the back of each sheet of drawing a minimum distance of 1.5 cm. (5/8 inch) down from the top of the page . . ." 37 C.F.R. § 1.84(c)).

(complete the following, if applicable)

☐ The enclosed drawing(s) are photograph(s), and there is also attached a "PETITION TO ACCEPT PHOTOGRAPH(S) AS DRAWING(S)." 37 C.F.R. § 1.84(b).

☐ formal

☒ informal

B. Other Papers Enclosed

0 Pages of declaration and power of attorney

1 Pages of abstract

0 Other

4. Additional papers enclosed

☐ Amendment to claims

☐ Cancel in this applications claims _____ before calculating the filing fee. (At least one original independent claim must be retained for filing purposes.)

☐ Add the claims shown on the attached amendment. (Claims added have been numbered consecutively following the highest numbered original claims.)

☐ Preliminary Amendment

☐ Information Disclosure Statement (37 C.F.R. § 1.98)

☐ Form PTO–1449 (PTO/SB/08A and 08B)

☐ Citations

☐ Declaration of Biological Deposit

☐ Submission of "Sequence Listing," computer readable copy and/or amendment pertaining thereto for biotechnology invention containing nucleotide and/or amino acid sequence.

☐ Authorization of Attorney(s) to Accept and Follow Instructions from Representative

☐ Special Comments

☐ Other

5. Declaration or oath (including power of attorney)

NOTE: A newly executed declaration is not required in a continuation or divisional application provided that the prior nonprovisional application contained a declaration as required, the application being filed is by all or fewer than all the inventors named in the prior application, there is no new matter in the application being filed, and a copy of the executed declaration filed in the prior application (showing the signature or an indication thereon that it was signed) is submitted. The copy must be accompanied by a statement requesting deletion of the names of person(s) who are not inventors of the application being filed. If the declaration in the prior application was filed under § 1.47, then a copy of that declaration must be filed accompanied by a copy of the decision granting § 1.47 status or, if a nonsigning person under § 1.47 has subsequently joined in a prior application, then a copy of the subsequently executed declaration must be filed. See 37 C.F.R. §§ 1.63(d)(1)–(3).

NOTE: A declaration filed to complete an application must be executed, identify the specification to which it is directed, identify each inventor by full name including family name and at least one given name, without abbreviation together with any other given name or initial, and the residence, post office address and country or citizenship of each inventor, and state whether the inventor is a sole or joint inventor. 37 C.F.R. § 1.63(a)(1)–(4).

NOTE: "The inventorship of a nonprovisional application is that inventorship set forth in the oath or declaration as prescribed by § 1.62, except as provided for in § 1.53(d)(4) and § 1.63(d). If an oath or declaration as prescribed by § 1.63 is not filed during the pendency of a nonprovisional application, the inventorship is that inventorship set forth in the application papers filed pursuant to § 1.53(b), unless a petition under this paragraph accompanied by the fee set forth in § 1.17(i) is filed supplying or changing the name or names of the inventor or inventors." 37 C.F.R. § 1.41(a)(1).

☐ Enclosed

Executed by

(check **all** *applicable boxes)*

☐ inventor(s).

☐ legal representative of inventor(s). 37 C.F.R. §§ 1.42 or 1.43.

☐ joint inventor or person showing a proprietary interest on behalf of inventor who refused to sign or cannot be reached.

☐ This is the petition required by 37 C.F.R. § 1.47 and the statement required by 37 C.F.R. § 1.47 is also attached. *See* item 13 below for fee.

☒ Not Enclosed.

NOTE: Where the filing is a completion in the U.S. of an International Application or where the completion of the U.S. application contains subject matter in addition to the International Application, the application may be treated as a continuation or continuation-in-part, as the case may be, utilizing ADDED PAGE FOR NEW APPLICATION TRANSMITTAL WHERE BENEFIT OF PRIOR U.S. APPLICATION CLAIMED.

☐ Application is made by a person authorized under 37 C.F.R. § 1.41(c) on behalf of *all* the above named inventor(s).

(The declaration or oath, along with the surcharge required by 37 C.F.R. § 1.16(e) can be filed subsequently).

☐ Showing that the filing is authorized.
(not required unless called into question. 37 C.F.R. § 1.41(d))

6. Inventorship Statement

WARNING: *If the named inventors are each not the inventors of all the claims an explanation, including the ownership of the various claims at the time the last claimed invention was made, should be submitted.*

The inventorship for all the claims in this application are:

☒ The same.

or

☐ Not the same. An explanation, including the ownership of the various claims at the time the last claimed invention was made,

☐ is submitted.

☐ will be submitted.

7. Language

NOTE: An application including a signed oath or declaration may be filed in a language other than English. An English translation of the non-English language application and the processing fee of $130.00 required by 37 C.F.R. § 1.17(k) is required to be filed with the application, or within such time as may be set by the Office. 37 C.F.R. § 1.52(d).

☒ English

☐ Non-English

☐ The attached translation includes a statement that the translation is accurate. 37 C.F.R. § 1.52(d).

8. Assignment

☒ An assignment of the invention to SIGHTSOUND.COM

☐ is attached. A separate ☐ "COVER SHEET FOR ASSIGNMENT (DOCUMENT) ACCOMPANYING NEW PATENT APPLICATION" or ☐ FORM PTO 1595 is also attached.

☒ will follow.

NOTE: "If an assignment is submitted with a new application, send two separate letters-one for the application and one for the assignment." Notice of May 4, 1990 (1114 O.G. 77-78).

WARNING: *A newly executed "CERTIFICATE UNDER 37 C.F.R. § 3.73(b)" must be filed when a continuation-in-part application is filed by an assignee. Notice of April 30, 1993, 1150 O.G. 62-64.*

9. Certified Copy

Certified copy(ies) of application(s)

Country	Appln. No.	Filed
Country	Appln. No.	Filed
Country	Appln. No.	Filed

from which priority is claimed

☐ is (are) attached.

☐ will follow.

NOTE: The foreign application forming the basis for the claim for priority must be referred to in the oath or declaration. 37 C.F.R. § 1.55(a) and 1.63.

NOTE: This item is for any foreign priority for which the application being filed directly relates. If any parent U.S. application or International Application from which this application claims benefit under 35 U.S.C. § 120 is itself entitled to priority from a prior foreign application, then complete item 18 on the ADDED PAGES FOR NEW APPLICATION TRANSMITTAL WHERE BENEFIT OF PRIOR U.S. APPLICATION(S) CLAIMED.

10. Fee Calculation (37 C.F.R. § 1.16)

A. ☒ Regular application

CLAIMS AS FILED				
Number filed	Number Extra		Rate	Basic Fee 37 C.F.R. § 1.16(a)
				$~~690.00~~ 710.00
Total Claims (37 C.F.R. § 1.16(c))	2 − 20 = 0	×	$ 18.00	0.00
Independent Claims (37 C.F.R. § 1.16(b))	2 − 3 = 0	×	$ 78.00	0.00
Multiple dependent claim(s), if any (37 C.F.R. § 1.16(d))		+	$260.00	

☐ Amendment cancelling extra claims is enclosed.

☐ Amendment deleting multiple-dependencies is enclosed.

☐ Fee for extra claims is not being paid at this time.

NOTE: If the fees for extra claims are not paid on filing they must be paid or the claims cancelled by amendment, prior to the expiration of the time period set for response by the Patent and Trademark Office in any notice of fee deficiency. 37 C.F.R. § 1.16(d).

Filing Fee Calculation $ 710.00

B. ☐ Design application ($310.00—37 C.F.R. § 1.16(f))

Filing Fee Calculation $_____

C. ☐ Plant application
($480.00—37 C.F.R. § 1.16(g))

Filing fee calculation $_____

11. Small Entity Statement(s)

☐ Statement(s) that this is a filing by a small entity under 37 C.F.R. § 1.9 and 1.27 is (are) attached.

WARNING: *"Status as a small entity must be specifically established in each application or patent in which the status is available and desired. Status as a small entity in one application or patent does not affect any other application or patent, including applications or patents which are directly or indirectly dependent upon the application or patent in which the status has been established. The refiling of an application under § 1.53 as a continuation, division, or continuation-in-part (including a continued prosecution application under § 1.53(d)), or the filing of a reissue application requires a new determination as to continued entitlement to small entity status for the continuing or reissue application. A nonprovisional application claiming benefit under 35 U.S.C. § 119(e), 120, 121, or 365(c) of a prior application, or a reissue application may rely on a statement filed in the prior application or in the patent if the nonprovisional application or the reissue application includes a reference to the statement in the prior application or in the patent or includes a copy of the statement in the prior application or in the patent and status as a small entity is still proper and desired. The payment of the small entity basic statutory filing fee will be treated as such a reference for purposes of this section." 37 C.F.R. § 1.28(a)(2).*

WARNING: *"Small entity status must not be established when the person or persons signing the . . . statement can **unequivocally** make the required self-certification." M.P.E.P., § 509.03, 6th ed., rev. 2, July 1996 (emphasis added).*

(complete the following, if applicable)

☐ Status as a small entity was claimed in prior application

_____ / _____, filed on _____, from which benefit is being claimed for this application under:

35 U.S.C. § ☐ 119(e),
☐ 120,
☐ 121,
☐ 365(c),

and which status as a small entity is still proper and desired.

☐ A copy of the statement in the prior application is included.

Filing Fee Calculation (50% of **A, B** or **C** above)

$_____

NOTE: Any excess of the full fee paid will be refunded if small entitiy status is established and a refund request are filed within 2 months of the date of timely payment of a full fee. The two-month period is not extendable under § 1.136. 37 C.F.R. § 1.28(a).

12. Request for International-Type Search (37 C.F.R. § 1.104(d))

(complete, if applicable)

☐ Please prepare an international-type search report for this application at the time when national examination on the merits takes place.

13. Fee Payment Being Made at This Time

☐ Not Enclosed

☐ No filing fee is to be paid at this time.
(This and the surcharge required by 37 C.F.R. § 1.16(e) can be paid subsequently.)

☒ Enclosed

☒ Filing fee $ 710.00

☐ Recording assignment ($40.00; 37 C.F.R. § 1.21(h)) (See attached "COVER SHEET FOR ASSIGNMENT ACCOMPANYING NEW APPLICATION".) $ _____

☐ Petition fee for filing by other than all the inventors or person on behalf of the inventor where inventor refused to sign or cannot be reached ($130.00; 37 C.F.R. §§ 1.47 and 1.17(i)) $ _____

☐ For processing an application with a specification in a non-English language ($130.00; 37 C.F.R. §§ 1.52(d) and 1.17(k)) $ _____

☐ Processing and retention fee ($130.00; 37 C.F.R. §§ 1.53(d) and 1.21(l)) $ _____

☐ Fee for international-type search report ($40.00; 37 C.F.R. § 1.21(e)) $ _____

NOTE: 37 C.F.R. § 1.21(l) establishes a fee for processing and retaining any application that is abandoned for failing to complete the application pursuant to 37 C.F.R. § 1.53(f) and this, as well as the changes to 37 C.F.R. §§ 1.53 and 1.78(a)(1), indicate that in order to obtain the benefit of a prior U.S. application, either the basic filing fee must be paid, or the processing and retention fee of § 1.21(l) must be paid, within 1 year from notification under § 53(f).

Total fees enclosed $ 710.00

14. Method of Payment of Fees

☒ Check in the amount of $ 710.00

☐ Charge Account No. _____ in the amount of $_____.

A duplicate of this transmittal is attached.

NOTE: Fees should be itemized in such a manner that it is clear for which purpose the fees are paid. 37 C.F.R. § 1.22(b).

15. Authorization to Charge Additional Fees

WARNING: *If no fees are to be paid on filing, the following items should not be completed.*

WARNING: *Accurately count claims, especially multiple dependent claims, to avoid unexpected high charges, if extra claim charges are authorized.*

☒ The Commissioner is hereby authorized to charge the following additional fees by this paper and during the entire pendency of this application to Account No. 19-0737 :

☒ 37 C.F.R. § 1.16(a), (f) or (g) (filing fees)

☒ 37 C.F.R. § 1.16(b), (c) and (d) (presentation of extra claims)

NOTE: *Because additional fees for excess or multiple dependent claims not paid on filing or on later presentation must only be paid or these claims cancelled by amendment prior to the expiration of the time period set for response by the PTO in any notice of fee deficiency (37 C.F.R. § 1.16(d)), it might be best not to authorize the PTO to charge additional claim fees, except possibly when dealing with amendments after final action.*

☐ 37 C.F.R. § 1.16(e) (surcharge for filing the basic filing fee and/or declaration on a date later than the filing date of the application)

☐ 37 C.F.R. § 1.17(a)(1)–(5) (extension fees pursuant to § 1.136(a)).

☐ 37 C.F.R. § 1.17 (application processing fees)

NOTE: *". . .A written request may be submitted in an application that is an authorization to treat any concurrent or future reply, requiring a petition for an extension of time under this paragraph for its timely submission, as incorporating a petition for extension of time for the appropriate length of time. An authorization to charge all required fees, fees under § 1.17, or all required extension of time fees will be treated as a constructive petition for an extension of time in any concurrent or future reply requiring a petition for an extension of time under this paragraph for its timely submission. Submission of the fee set forth in § 1.17(a) will also be treated as a constructive petition for an extension of time in any concurrent reply requiring a petition for an extension of time under this paragraph for its timely submission." 37 C.F.R. § 1.136(a)(3).*

☐ 37 C.F.R. § 1.18 (issue fee at or before mailing of Notice of Allowance, pursuant to 37 C.F.R. § 1.311(b))

NOTE: *Where an authorization to charge the issue fee to a deposit account has been filed before the mailing of a Notice of Allowance, the issue fee will be automatically charged to the deposit account at the time of mailing the notice of allowance. 37 C.F.R. § 1.311(b).*

NOTE: *37 C.F.R. § 1.28(b) requires "Notification of any change in status resulting in loss of entitlement to small entity status must be filed in the application . . . prior to paying, or at the time of paying, . . . the issue fee. . . " From the wording of 37 C.F.R. § 1.28(b), (a) notification of change of status must be made even if the fee is paid as "other than a small entity" and (b) no notification is required if the change is to another small entity.*

16. Instructions as to Overpayment

NOTE: ". . . Amounts of twenty-five dollars or less will not be returned unless specifically requested within a reasonable time, nor will the payer be notified of such amounts; amounts over twenty-five dollars may be returned by check or, if requested, by credit to a deposit account." 37 C.F.R. § 1.26(a).

☒ Credit Account No. 19-0737

☐ Refund

SIGNATURE OF PRACTITIONER

Reg. No. 30,587

Ansel M. Schwartz

(type or print name of attorney)

Tel. No. (412) 621-9222

One Sterling Plaza
201 N. Craig Street, Suite 304

P.O. Address

Customer No.

Pittsburgh, PA 15213

☐ **Incorporation by reference of added pages**

(check the following item if the application in this transmittal claims the benefit of prior U.S. application(s) (including an international application entering the U.S. stage as a continuation, divisional or C-I-P application) and complete and attach the ADDED PAGES FOR NEW APPLICATION TRANSMITTAL WHERE BENEFIT OF PRIOR U.S. APPLICATION(S) CLAIMED)

☐ Plus Added Pages for New Application Transmittal Where Benefit of Prior U.S. Application(s) Claimed

Number of pages added _____

☐ Plus Added Pages for Papers Referred to in Item 4 Above

Number of pages added _____

☐ Plus added pages deleting names of inventor(s) named in prior application(s) who is/are no longer inventor(s) of the subject matter claimed in this application.

Number of pages added _____

☐ Plus "Assignment Cover Letter Accompanying New Application"

Number of pages added _____

☒ **Statement Where No Further Pages Added**

(if no further pages form a part of this Transmittal, then end this Transmittal with this page and check the following item)

☒ This transmittal ends with this page.

METHOD AND SYSTEM FOR ESTABLISHING A TRUSTED AND DECENTRALIZED PEER-TO-PEER NETWORK

FIELD OF THE INVENTION

The present invention is related to a trusted and decentralized peer-to-peer network method and system.

BACKGROUND OF THE INVENTION

Important to the business of selling movies electronically via the Internet includes efficient electronic distribution and/or file sharing. The process of sharing computer files (including digitized movies) via communications means has increased in importance with the proliferation of the Internet for electronic distribution and file transfer. The creation of file sharing and/or peer-to-peer networks (e.g. Gnutella) has enabled users of computer workstations, which lack traditional serving software to distribute computer files. Traditionally, serving computers (e.g. computers running Microsoft Windows 2000 Server) performed the task of distributing computer files to client work stations using a centralized network architecture. The advent of file sharing and/or peer-to-peer networks gave way to a decentralized network architecture comprised of multiple computer workstations (e.g. host nodes) acting as redundant repositories, each capable of transferring the same computer files. Some of the current file sharing and/or peer-to-peer networks communicate in an open and un-trusted manner. Additionally, trusted peer-to-peer networks have relied on a centralized process of identifying members and their related IP address to establish the trusted peer-to-peer network.

SUMMARY OF THE INVENTION

The present invention offers a new and improved method and system to establish a trusted and decentralized peer-to-peer network for: the sharing of computer files between and among computing devices; trusted chat sessions; and for other applications of trusted peer-to-peer networks. Additionally, the present invention also offers a new and improved method and system to provide file identification properties or attributes prior to the actual download of the file through file sharing utilizing a trusted and decentralized peer-to-peer network. The present invention can be a software program residing computing devices permitting users to automatically interact in a trusted peer-to-peer manner during the file sharing process.

BRIEF DESCRIPTION OF THE DRAWINGS

In the accompanying drawings, the preferred embodiment of the invention and preferred methods of practicing the invention are illustrated in which:

Fig. 1 is a schematic diagram which may be used in carrying out the teachings of this invention for the purpose of establishing a trusted and decentralized peer-to-peer network to depict a configuration in which computing devices can be connected to in conjunction with the establishment of a trusted and decentralized peer-to-peer network.

Fig. 2 is a computer programming flowchart.

Fig. 3 is a computer programming flowchart.

Fig. 4 is a computer programming flowchart.

Fig. 5 is a computer programming flowchart.

Fig. 6 is a computer programming flowchart.

Fig. 7 is a computer programming flowchart.

Fig. 8 is a computer programming flowchart.

Fig. 9 is a computer programming flowchart.

DETAILED DESCRIPTION

Referring now to the drawings wherein like reference numerals refer to similar or identical parts throughout the several views, and more specifically to figure 1 thereof, there is shown a system for establishing a trusted and decentralized peer-to-peer network. The system comprises multiple computing devices (e.g. Device "A" **10**, Device "B" **11**) each having a connection to a Communications Means **20** and each possessing the means to: utilize communication protocols (e.g. FTP Protocol **150**, HTTP Protocol **151**, Chat Protocol **152**, File Sharing Protocols **153**); utilize a communications program (e.g. the E-mail Program **130**); transfer or download the computer files. The system comprises multiple computing devices (e.g. Device "A" **10**, Device "B" **11**) each having a User Interface **20** through which the user of the respective computing devices interfaces. The system comprises multiple computing devices (e.g. Device "A" **10**, Device "B" **11**) each having a Processor **21** that processes computation instructions. The system

comprises multiple computing devices (e.g. Device "A" **10**, Device "B" **11**) each having a RAM **22** that provides memory for the respective computing devices. The system comprises multiple computing devices (e.g. Device "A" **10**, Device "B" **11**) each having a Storage **23** that provides persistent memory or storage for the respective computing devices. The system comprises multiple computing devices (e.g. Device "A" **10**, Device "B" **11**) each having a Transceiver **24** that connects the respective computing devices to the Communications Means **30** and through which communications are transferred between the computing devices. The Device "A" **10** is separate, apart and distinct from the Device "B" **11**.

Preferably, the Peer-to-Peer Network Program **40** is connected to a User Interface **20** of a computing device (e.g. the Device "A" **10**) which enables the user of the Device "A" **10** to input information to the Peer-to-Peer Network Program **40**. The Peer-to-Peer Network Program **40** of a computing device (e.g. the Device "A" **10**) can interact with the Peer-to-Peer Network Program **40** of another computing device (e.g. the Device "B" **11**).

The present invention pertains to a method to establish a trusted and decentralized peer-to-peer network. The method comprises the step of initially installing the Peer-to-Peer Network Program **40**. Then there is preferably the step of creating encryption and decryption keys through means of a CryptoAPI **70** of an Operating System **25**. Then there is preferably the step of creating a searchable ciphertext file containing identifiable network information on each computing device embodied by the present invention, which can be shared with each of the other such

computing devices. Then there is preferably the step of appending data and/or other information to, or associating data and/or other information with, a specific computer file to be included in the file sharing functionality of the trusted and decentralized peer-to-peer network created by the present invention. Then there is preferably the step of distributing public keys (such as, but not limited to, E-mail, CD-ROM, etc.) from one computing device to the computing devices of other members that belong to a given trusted network, and preparing such public keys for use in the decryption of encrypted files between the members of the trusted peer-to-peer network. Then there is preferably the step of whereby a member of the trusted peer-to-peer network can find other members of the trusted peer-to-peer network through decentralized means. Then there is preferably the step of using the trusted peer-to-peer network for various communications purposes such as, but not limited to: trusted chat sessions, trusted file sharing, etc.

Referring now to the drawings wherein like reference numerals refer to similar or identical parts throughout the several views, and more specifically to **FIG. 1** through **FIG. 9** thereof, there is shown an apparatus **40** for invoking functionality of the Operating System **25** of computing devices Device "A" **10** and Device "B" **11**. The apparatus **40** is connected to the Operating System **25** of computing devices Device "A" **10** and Device "B" **11**. The apparatus **40** comprises means for invoking functionality of an Operating System **25** of a computing devices the Device "A" **10** to coordinate with the apparatus **40** of another computing device the Device "B" **11** to: share decryption keys (e.g. UserA Public Key **80**, UserB Public Key **81**) via electronic or manual means; share encrypted "FindMe" files (e.g. UserA FindMe File **100**, UserB FindMe

File **101**) via open and un-trusted file sharing networks; establish a trusted peer-to-peer network between computing devices Device "A" **10** and Device "B" **11**; real-time location of members of the trusted peer-to-peer network; communication between and among computing devices comprising the trusted peer-to-peer network; file sharing between and among computing devices comprising the trusted peer-to-peer network.

Fig. 1 is a schematic diagram which may be used in carrying out the teachings of this invention for the purpose of establishing a trusted and decentralized peer-to-peer network to depict a configuration in which computing devices can be connected to in conjunction with the establishment of a trusted and decentralized peer-to-peer network.

Fig. 2 is a computer programming flowchart which may be used in carrying out the teachings of this invention for the purpose of installing software (e.g. the Peer-to-Peer Network Program **40**) which is capable of executing all, or a part, of the teachings of this invention.

Fig. 3 is a computer programming flowchart which may be used in carrying out the teachings of this invention depicting how the Peer-to-Peer Network Program **40** can be designed to automatically invoke functionality an operating system (e.g. the Operating System **25**) to create encryption and decryption keys.

Fig. 4 is a computer programming flowchart which may be used in carrying out the teachings of this invention depicting how the Peer-to-Peer Network Program **40** can be designed to automatically: create searchable ciphertext files containing the

information necessary to create a decentralized control procedure for the creation of the trusted peer-to-peer network; permit the user of the computing device to manually input user information; insert into the searchable files peer-to-peer network information derived from an operating system (e.g. the Operating System **25**) and insert user information; encrypt the searchable files using encryption keys (see Fig. 3); and saving the encrypted and searchable ciphertext file to a searchable file folder on a storage device (e.g. the Storage **23**).

Fig. 5 is a computer programming flowchart which may be used in carrying out the teachings of this invention depicting how the Peer-to-Peer Network Program **40** can be designed to automatically append data and/or information (e.g. file attributes, file properties, etc.) to a computer file.

Fig. 6 is a computer programming flowchart which may be used in carrying out the teachings of this invention depicting how the Peer-to-Peer Network Program **40** can assist the user of a computing device to automatically distribute decryption keys (e.g. the public keys) to user specified recipients (e.g. members of the trusted peer-to-peer network).

Fig. 7 is a computer programming flowchart which may be used in carrying out the teachings of this invention depicting how the Peer-to-Peer Network Program **40** can automatically: search an open (non-trusted) file sharing network for computer files (e.g. UserA FindMe File **100**) containing network information (e.g. IP address) pertaining to computing devices controlled or operated by specific and previously known users (e.g. members of the trusted peer-to-peer network); download a copy of one or more of the

computer files (e.g. UserA FindMe File **100**); decrypt the computer files (e.g. UserA FindMe File **100**); extract the network information (e.g. IP address); and associate the network information (e.g. IP address) with the specific and previously known users (e.g. members of the trusted peer-to-peer network) respectively in a trusted member list (e.g. the Trusted Member List **140**).

Fig. 8 is a computer programming flowchart which may be used in carrying out the teachings of this invention depicting how the Peer-to-Peer Network Program **40** can automatically invoke the functionality of communications protocols (e.g. the Chat Protocol **152**) to establish a chat session with specific and previously known users (e.g. members of the trusted peer-to-peer network).

Fig. 9 is a computer programming flowchart which may be used in carrying out the teachings of this invention depicting how the Peer-to-Peer Network Program **40** can automatically: establish a trusted peer-to-peer network using a the IP addresses in the Trusted Member List **140**; invoke the functionality of communications protocols (e.g. the File Sharing Protocols **153**) to execute computer file searches on, and retrieval from, computing devices controlled or operated by users listed on the Trusted Member List **140**.

Referring now to **FIG. 1** through **FIG. 9**, a preferred embodiment of the invention is comprised of the following:

10 Device "A"

11 Device "B"

12 Device "C"

13	Device "D"
20	User Interface
21	Processor
22	RAM
23	Storage
24	Transceiver
25	Operating System
30	Communications Means
41	Setup.exe File
40	Peer-to-Peer Network Program 40
50	User Information File
60	FindMe Folder
61	Searched FindMe Results Folder
62	Trusted Search Folder
70	CryptoAPI
71	Cryptographic Service Provider
72	Application Programming Interface
80	UserA Public Key
81	UserB Public Key
82	UserC Public Key
83	UserD Public Key
90	UserA Private Key
91	UserB Private Key
92	UserC Private Key
93	UserD Private Key
100	UserA FindMe File
101	UserB FindMe File

102 UserC FindMe File

103 UserD FindMe File

110 UserA Public Key BLOB

111 UserB Public Key BLOB

112 UserC Public Key BLOB

113 UserD Public Key BLOB

120 File Info Stream

130 E-mail Program

140 Trusted Member List

150 FTP Protocol

151 HTTP Protocol

152 Chat Protocol

153 File Sharing Protocols

In **FIG. 1** through **FIG. 9**, the following components are already commercially available: the Device "A" **10**, the Device "B" **11**, the Device "C" **12**, the Device "D" **13**, the User Interface **20**, the Processor **21**, the RAM **22**, the Storage **23**, the Transceiver **24**, the Operating System **25**, the Communications Means **30**, the CryptoAPI **70**, the Cryptographic Service Provider **71**, the Application Programming Interface **72**, the UserA Public Key **80**, the UserB Public Key **81**, the UserC Public Key **82**, the UserD Public Key **83**, the UserA Private Key **90**, the UserB Private Key **91**, the UserC Private Key **92**, the UserD Private Key **93**, the UserA Public Key BLOB **110**, the UserB Public Key BLOB **111**, the UserC Public Key BLOB **112**, the UserD Public Key BLOB **113**, the E-mail Program **130**, the FTP Protocol **150**, the HTTP Protocol **151**, the Chat Protocol **152**, and the File Sharing

Protocols **153**. The Peer-to-Peer Network Program **40**, the Setup.exe File **41**, the User Information File **50**, the FindMe Folder **60**, the Searched FindMe Results Folder **61**, the Trusted Search Folder **62**, the UserA FindMe File **100**, the UserB FindMe File **101**, the UserC FindMe File **102**, the UserD FindMe File **103**, the File Info Stream **120**, and the Trusted Member List **140**, are new teachings of this invention.

The Device "A" **10**, the Device "B" **11**, the Device "C" **12**, the Device "D" **13** are means, which can perform computational functions (such as, but not limited to, a Web Server, PC, Mac, PalmPC, Laptop, etc.). The Device "A" **10**, the Device "B" **11**, the Device "C" **12**, the Device "D" **13** are also means, which can electronically communicate with other computing devices. The Device "A" **10**, the Device "B" **11**, the Device "C" **12**, the Device "D" **13** are also means to transmit computer files (e.g. the UserA FindMe File **100**) to other computing devices. The Device "A" **10**, the Device "B" **11**, the Device "C" **12**, the Device "D" **13** are also means to transmit information and/or requests to other computing devices. The Device "A" **10**, the Device "B" **11**, the Device "C" **12**, the Device "D" **13** may contain a video display, audio speakers, and other computing peripherals. The present invention is not limited to only the Device "A" **10**, the Device "B" **11**, the Device "C" **12**, and the Device "D" **13**.

The User Interface **20** (such as, but not limited to, web browser software such as the Microsoft Internet Explorer, keyboard, mouse, video monitor, speakers) is means, which the user of a computer device (e.g. Device "A" **10**) can interface with the

computing device. The User Interface **20** is means, which can be used by the user of the computing device to transmit requests to another computing device and can display the contents of the User Interface **20** to the user of the computing device. The User Interface **20** is means, which can receive and execute requests transmitted from another computing device. The User Interface **20** is also means, which is a client program that can use the hypertext transfer protocol ("HTTP") to make requests of a plurality of devices (e.g. Device "A" **10**) throughout the Internet on behalf of the user of any other device of devices (e.g. Device "C" **12**).

The Processor **21** is means of a computing device (e.g. Device "A" **10**, Device "B" **11**, Device "C" **12**, and Device "D" **13**) to electronically process instructions of the Operating System **25**, other computer programs running on the Operating System **25** or other computer peripheral devices of the computing device (e.g. Device "A" **10**, Device "B" **11**, Device "C" **12**, and Device "D" **13**). The Processor **21** is also means of a computing device (e.g. Device "A" **10**, Device "B" **11**, Device "C" **12**, and Device "D" **13**) to electronically process instructions of other peripheral software and/or firmware devices of the a computing device.

The RAM **22** is means used by the Operating System **25** of a computing device (e.g. Device "A" **10**, Device "B" **11**, Device "C" **12**, and Device "D" **13**) to temporarily store computer files, computer programs or other computer information (such as, but not limited to, the UserA FindMe File **100**) for use by the Operating System **25**, computer programs running on the Operating System **25** or other computer peripheral devices of the computing devices.

The Storage **23** is means in, or connected to, a computing device (e.g. Device "A" **10**, Device "B" **11**, Device "C" **12**, and Device "D" **13**), which can be used to electronically save an electronic copy of the digital code of a computer program or computer file (such as, but not limited to, the UserA FindMe File **100**) from RAM **22** of the a computing device.

The Transceiver **24** (such as, but not limited to. a telephone modem, cable modem, network interface card, etc.) is means to electronically send and receive communication signals via a Communications Means **30**. The Transceiver **24** is means used by software and/or firmware of, or connected to, a computing device (e.g. Device "A" **10**, Device "B" **11**, Device "C" **12**, and Device "D" **13**) and/or the Operating System **25** thereof, to electronically communicate via a Communications Means **30**. The Transceiver **24** is connected to a computing device (e.g. Device "A" **10**, Device "B" **11**, Device "C" **12**, and Device "D" **13**) and is connected to the Communications Means **30**.

The Operating System **25** (such as, but not limited to, Microsoft Windows 2000) is means to permit computing functionality of a computing device (e.g. Device "A" **10**, Device "B" **11**, Device "C" **12**, and Device "D" **13**).

The Communications Means **30** (such as, but not limited to, telephone lines, cable TV lines, coax cable, fiber optics, radio, cellular, satellite, serial cables, parallel cables, infrared communication, laser communication, universal serial bus (USB) cables, the Internet, LAN, Ethernet, network generally, etc.) is

means by which computing devices connected thereto can electronically communicate. The Communications Means **30** is also means by which computing devices connected thereto can invoke standard communication protocols (such as, but not limited to, hypertext transfer protocol (HTTP); file transfer protocol (FTP); etc.) to transmit and receive signals and/or computer programs or computer files (such as, but not limited to, the UserA FindMe File **100**). The Communications Means **30** is also means by which computing devices connected thereto can invoke encrypted communication protocols (such as, but not limited to, secure sockets layer (SSL), transport layer security (TLS), virtual private network (VPN), etc.) to transmit and receive encrypted signals. The Communications Means **30** is also means which can include a worldwide system of computer networks, or a network of networks, known as the "Internet" in which users at any one computing device can get information from any other computer device. The Communications Means **30** is connected to the Transceiver **70**, a computing device (e.g. Device "A" **10**, Device "B" **11**, Device "C" **12**, and Device "D" **13**).

The Peer-to-Peer Network Program **40** is means, which can operate on a plurality of computing devices (e.g. Device "A" **10**, Device "B" **11**, Device "C" **12**, and Device "D" **13**). The Peer-to-Peer Network Program **40** is also means to instruct the Operating System **25**, or a communication program thereof, to communicate with another computing device (e.g. Device "C" **12**) via Communications Means **30**. The Peer-to-Peer Network Program **40** of a computing device (e.g. Device "D" **13**) is also means to instruct the Peer-to-Peer Network Program **40** of a computing device (e.g. Device "A" **10**) to transmit

queries or instructions to the Operating System **25** of the Device "A" **10** during the execution of the functionality of the Peer-to-Peer Network Program **40** of the Device "A" **10** and the Device "D" **13**. The Peer-to-Peer Network Program **40** of a computing device (e.g. Device "D" **13**) is also means to receive instructions from the Peer-to-Peer Network Program **40** of a computing device (e.g. Device "A" **10**) via Communications Means **30**. The Peer-to-Peer Network Program **40** of a computing device (e.g. Device "A" **10**) is also means to automatically receive from transmission, a computer file (e.g. the UserC FindMe File **102**) transmitted from another computer device (e.g. Device "C" **12**) via a Communications Means **30** and place an electronic copy thereof in RAM **22** on the first computing device (e.g. Device "A" **10**). The Peer-to-Peer Network Program **40** is also means to automatically instruct the Operating System **25** of a computing device (e.g. Device "A" **10**) to recall a computer file (e.g. the UserC FindMe File **102**) from RAM **22** and save an electronic copy thereof to Storage **23**. The Peer-to-Peer Network Program **40** is means to transmit a message to the User Interface **20** of a computing device (e.g. Device "A" **10**) upon completion of the execution of the functionality of the Peer-to-Peer Network Program **40**. The Peer-to-Peer Network Program **40** is also means to enable users of computing devices (e.g. Device "A" **10**, Device "B" **11**, Device "C" **12**, and Device "D" **13**) to establish a trusted communications network. The Peer-to-Peer Network Program **40** is also means to enable users of computing devices (e.g. Device "A" **10**, Device "B" **11**, Device "C" **12**, and Device "D" **13**) to utilize the trusted communications network to establish a communications session (e.g. a chat

session). The Peer-to-Peer Network Program **40** is also means to enable users of computing devices (e.g. Device "A" **10**, Device "B" **11**, Device "C" **12**, and Device "D" **13**) to utilize the trusted communications network to share files between the computing devices. The Peer-to-Peer Network Program **40** is also means to utilize communications protocols (e.g. the FTP Protocol **150**) to automatically transmit a computer file (e.g. the UserA FindMe File **100**) to other computing devices, operating with the Peer-to-Peer Network Program **40**, via Communications Means **30**. The Peer-to-Peer Network Program 40 may be embodied in computer coding software (such as, but not limited to, a program authored in the computer language C++, C#, Active Server Pages, XML, Visual Basic, ActiveX Controls, Java Script, etc.) to execute the described functions.

The Setup.exe File **41** is means, which can be transmitted to a computing device (e.g. Device "A" **10**, Device "B" **11**, Device "C" **12**, and Device "D" **13**) via various means (e.g. via download, CD-ROM, etc.). The Setup.exe File **41** is means, which functionality can be invoked by a user of a computing device by a common process of "double clicking" on the file itself. The Setup.exe File **41** is means, which contains a copy of the Peer-to-Peer Network Program **40**. The Setup.exe File **41** is means, which automatically installs the Peer-to-Peer Network Program **40** on a computing device (e.g. Device "A" **10**, Device "B" **11**, Device "C" **12**, and Device "D" **13**) and makes the necessary modifications to the Operating System **25** of the computing device necessary for proper functioning of the Peer-to-Peer Network Program **40**.

The User Information File **50** is a computer file (such as, but not limited to, a text document, etc.) which is created by the Peer-to-Peer Network Program **40** from information manually inputted by the user of the computing device (e.g. Device "A" **10**, Device "B" **11**, Device "C" **12**, and Device "D" **13**) running the copy of the Peer-to-Peer Network Program **40**. The User Information File **50** is also a computer file, which can include information specific to the host computing device, which the information can be automatically collected by the Peer-to-Peer Network Program **40** utilizing the Application Programming Interface **72** of the host Operating System **25**.

The FindMe Folder **60** is the folder location in the Storage **23** of the host computing device (e.g. Device "B" **11**) where a searchable file (e.g. UserB FindMe File **101**) containing trusted network information on the computing device, and the user thereof, is to be saved. The FindMe Folder **60** is created by the Setup.exe File **41** upon initial installation of the Peer-to-Peer Network Program **40**. The FindMe Folder **60** can also be created by the Peer-to-Peer Network Program **40** after the initial installation of the Peer-to-Peer Network Program **40**, either automatically or upon request by the respective user. Multiple FindMe Folders **60** can be present on each computing device containing an embodiment of this invention.

The Searched FindMe Results Folder **61** is the folder location in the Storage **23** of the host computing device (e.g. Device "B" **11**) where computer files (e.g. UserA FindMe File **100**,

UserC FindMe File **102**, UserD FindMe File **103**) resulting from specific search queries performed by the Peer-to-Peer Network Program **40** are to be saved by the Peer-to-Peer Network Program **40**. The Searched FindMe Results Folder **61** is created by the Setup.exe File **41** upon initial installation of the Peer-to-Peer Network Program **40**. The Searched FindMe Results Folder **61** can also be created by the Peer-to-Peer Network Program **40** after the initial installation of the Peer-to-Peer Network Program **40**, either automatically or upon request by the respective user. Multiple Searched FindMe Results Folders **61** can be present on each computing device containing an embodiment of this invention.

The Trusted Search Folder **62** is the folder location in the Storage **23** of the host computing device (e.g. Device "B" **11**) where computer files (e.g. documents, spreadsheets, audio files (e.g. ASF, WMA, MP3, WAV, AUI), video files (e.g. ASF, WMV, AVI, MPEG), executable programs (e.g. EXE), etc.) resulting from specific search queries performed by the Peer-to-Peer Network Program **40** are to be saved by the Peer-to-Peer Network Program **40**. The Trusted Search Folder **62** can be searched by other computing devices running a copy of the Peer-to-Peer Network Program **40**. The Trusted Search Folder **62** can be searched by other computing devices running a copy of the Peer-to-Peer Network Program **40** utilizing File Sharing Protocols **153**. The Trusted Search Folder **62** is created by the Setup.exe File **41** upon initial installation of the Peer-to-Peer Network Program **40**. The Trusted Search Folder **62** can also be created by the Peer-to-Peer Network Program **40** after the initial installation of the Peer-to-Peer Network Program **40**, either automatically or upon request by the respective user. Multiple

Trusted Search Folders **62** can be present on each computing device containing an embodiment of this invention.

The CryptoAPI **70** is means of an operating system (e.g. the Operating System **25**), which enables computer programs and/or applications to execute cryptographic functions of the operating system (e.g. the Microsoft® CryptoAPI, an application programming interface). The CryptoAPI **70** is also means of an Operating System **25**, which contains cryptographic functionality and which the Peer-to-Peer Network Program **40** can utilize to execute encrypting and decrypting functions. The CryptoAPI **70** is also means of an Operating System **25**, which includes functionality for encrypting and decrypting data, and for authentication using digital certificates. The CryptoAPI **70** is also means of an Operating System **25**, which provides certificate management functions maintaining and managing a persistent storage of certificates, public keys, and private keys in a certificate store (e.g. the Cryptographic Service Provider **71**). The CryptoAPI **70** is also means of an Operating System **25**, which contains functions that can incorporate certificates in outgoing transmissions and/or messages and verify digital certificates that are being received in received transmissions and/or messages.

The Cryptographic Service Provider **71** is means of the CryptoAPI **70**, which stores public/private key pairs from session to session in persistent memory (e.g. the Microsoft® CSP). The Cryptographic Service Provider **71** is also means of the CryptoAPI **70**, which can store public/private key pairs, in encrypted form, in

the system registry of the operating system (e.g. the Operating System **25**).

The Application Programming Interface **72** is means of an operating system (e.g. the Operating System **25**), which enables the Peer-to-Peer Network Program **40** to programmatically retrieve specific information about the host computer (e.g. the Microsoft® Win32 API). The Application Programming Interface **72** is also means of an operating system (e.g. the Operating System **25**), which enables the Peer-to-Peer Network Program **40** to programmatically retrieve specific IP address information about the network configuration of the host computer. The Application Programming Interface **72** may include functionality of the CryptoAPI **70**, the Cryptographic Service Provider **71**, and other similar application programming interfaces.

The UserA Public Key **80** is means, which is used to decrypt files that have been encrypted with that certain private key (e.g. UserA Private Key **90**) which forms the public/private key pair with the UserA Public Key **80**. The UserA Public Key **80** is also means, which is created by the Peer-to-Peer Network Program **40** using the CryptoAPI **70**. The UserA Public Key **80** is also means, which can be stored in persistent memory on the Storage **23** of the host computing device (e.g. Device "A" **10**) by the Cryptographic Service Provider **71**. The UserA Public Key **80** is also means, which can be transmitted to another computing device (e.g. Device "B" **11**, Device "C" **12**, and Device "D" **13**) via various means (e.g. via Communications Means **30**, floppy disk, E-mail Program **130**, etc.).

The UserA Public Key **80** is also means, which can be stored in persistent memory on the Storage **23** of other computing device (e.g. Device "B" **11**, Device "C" **12**, and Device "D" **13**) by the Cryptographic Service Provider **71**. The UserA Public Key **80** is also means, which can decrypt files on other computing device (e.g. Device "B" **11**, Device "C" **12**, and Device "D" **13**), that have been encrypted with that certain private key (e.g. UserA Private Key **90**).

The UserB Public Key **81** is means, which is used to decrypt files that have been encrypted with that certain private key (e.g. UserB Private Key **91**) which forms the public/private key pair with the UserB Public Key **81**. The UserB Public Key **81** is also means, which is created by the Peer-to-Peer Network Program **40** using the CryptoAPI **70**. The UserB Public Key **81** is also means, which can be stored in persistent memory on the Storage **23** of the host computing device (e.g. Device "B" **11**) by the Cryptographic Service Provider **71**. The UserB Public Key **81** is also means, which can be transmitted to another computing device (e.g. Device "A" **10**, Device "C" **12**, and Device "D" **13**) via various means (e.g. via Communications Means **30**, floppy disk, E-mail Program **130**, etc.). The UserB Public Key **81** is also means, which can be stored in persistent memory on the Storage **23** of other computing device (e.g. Device "A" **10**, Device "C" **12**, and Device "D" **13**) by the Cryptographic Service Provider **71**. The UserB Public Key **81** is also means, which can decrypt files on other computing device (e.g. Device "A" **10**, Device "C" **12**, and Device "D" **13**), that have been

encrypted with that certain private key (e.g. UserB Private Key **91**).

The UserC Public Key **82** is means, which is used to decrypt files that have been encrypted with that certain private key (e.g. UserC Private Key **92**) which forms the public/private key pair with the UserC Public Key **82**. The UserC Public Key **82** is also means, which is created by the Peer-to-Peer Network Program **40** using the CryptoAPI **70**. The UserC Public Key **82** is also means, which can be stored in persistent memory on the Storage **23** of the host computing device (e.g. Device "C" **12**) by the Cryptographic Service Provider **71**. The UserC Public Key **82** is also means, which can be transmitted to another computing device (e.g. Device "A" **10**, Device "B" **11**, and Device "D" **13**) via various means (e.g. via Communications Means **30**, floppy disk, E-mail Program **130**, etc.). The UserC Public Key **82** is also means, which can be stored in persistent memory on the Storage **23** of other computing device (e.g. Device "A" **10**, Device "B" **11**, and Device "D" **13**) by the Cryptographic Service Provider **71**. The UserC Public Key **82** is also means, which can decrypt files on other computing device (e.g. Device "A" **10**, Device "B" **11**, and Device "D" **13**), that have been encrypted with that certain private key (e.g. UserC Private Key **92**).

The UserD Public Key **83** is means, which is used to decrypt files that have been encrypted with that certain private key (e.g. UserD Private Key **93**) which forms the public/private key pair with the UserD Public Key **83**. The UserD Public Key **83** is also

means, which is created by the Peer-to-Peer Network Program **40** using the CryptoAPI **70**. The UserD Public Key **83** is also means, which can be stored in persistent memory on the Storage **23** of the host computing device (e.g. Device "D" **13**) by the Cryptographic Service Provider **71**. The UserD Public Key **83** is also means, which can be transmitted to another computing device (e.g. Device "A" **10**, Device "B" **11**, and Device "C" **12**) via various means (e.g. via Communications Means **30**, floppy disk, E-mail Program **130**, etc.). The UserD Public Key **83** is also means, which can be stored in persistent memory on the Storage **23** of other computing device (e.g. Device "A" **10**, Device "B" **11**, and Device "C" **12**) by the Cryptographic Service Provider **71**. The UserD Public Key **83** is also means, which can decrypt files on other computing device (e.g. Device "A" **10**, Device "B" **11**, and Device "C" **12**), that have been encrypted with that certain private key (e.g. UserD Private Key **93**).

The UserA Private Key **90** is means, which is used to encrypt files, which can be decrypted with that certain public key (e.g. UserA Public Key **80**) which forms the public/private key pair with the UserA Private Key **90**. The UserA Private Key **90** is also means, which is created by the Peer-to-Peer Network Program **40** using the CryptoAPI **70**. The UserA Private Key **90** is also means, which can be stored in persistent memory on the Storage **23** of the host computing device (e.g. Device "A" **10**) by the Cryptographic Service Provider **71**.

The UserB Private Key **91** is means, which is used to encrypt files, which can be decrypted with that certain public key (e.g. UserB Public Key **81**) which forms the public/private key pair with the UserB Private Key **91**. The UserB Private Key **91** is also means, which is created by the Peer-to-Peer Network Program **40** using the CryptoAPI **70**. The UserB Private Key **91** is also means, which can be stored in persistent memory on the Storage **23** of the host computing device (e.g. Device "B" **11**) by the Cryptographic Service Provider **71**.

The UserC Private Key **92** is means, which is used to encrypt files, which can be decrypted with that certain public key (e.g. UserC Public Key **82**) which forms the public/private key pair with the UserC Private Key **92**. The UserC Private Key **92** is also means, which is created by the Peer-to-Peer Network Program **40** using the CryptoAPI **70**. The UserC Private Key **92** is also means, which can be stored in persistent memory on the Storage **23** of the host computing device (e.g. Device "C" **12**) by the Cryptographic Service Provider **71**.

The UserD Private Key **93** is means, which is used to encrypt files, which can be decrypted with that certain public key (e.g. UserD Public Key **83**) which forms the public/private key pair with the UserD Private Key **93**. The UserD Private Key **93** is also means, which is created by the Peer-to-Peer Network Program **40** using the CryptoAPI **70**. The UserD Private Key **93** is also means, which can be stored in persistent memory on the Storage **23** of the

host computing device (e.g. Device "D" **13**) by the Cryptographic Service Provider **71**.

The UserA FindMe File **100** is a computer file (e.g. a text file), which is created by the Peer-to-Peer Network Program **40**. The UserA FindMe File **100** contains information relating to the Device "A" **10**, and the user thereof. The UserA FindMe File **100** can contain information such as: the system name of the Device "A" **10**; the name of the user of the Device "A" **10**; the IP address of the Device "A" **10**; etc. The UserA FindMe File **100** is created by the Peer-to-Peer Network Program **40** and can contain information from the User Information File **50**. The UserA FindMe File **100** is created by the Peer-to-Peer Network Program **40** and can contain information obtained by the Peer-to-Peer Network Program **40** through use of the Application Programming Interface **72**. The UserA FindMe File **100** can be created by the Peer-to-Peer Network Program **40** each time the Peer-to-Peer Network Program **40** is started, and any old versions of the UserA FindMe File **100** can be overwritten with a new version, to account for information changes to relating to the Device "A" **10**, and/or the user thereof. The UserA FindMe File **100** can exist in plaintext form and/or ciphertext form. The UserA FindMe File **100** can be encrypted by the Peer-to-Peer Network Program **40** with the UserA Private Key **90** through use of the CryptoAPI **70** of the Device "A" **10**. The UserA FindMe File **100** can be decrypted by the Peer-to-Peer Network Program **40** with the UserA Public Key **80** through use of the CryptoAPI **70** of the Device "A" **10**. The UserA FindMe File **100** can be decrypted by the Peer-to-Peer Network Program **40** with the

UserA Public Key **80** through use of the CryptoAPI **70** of other computing devices (e.g. Device "B" **11**, Device "C" **12**, and Device "D" **13**). The UserA FindMe File **100** can be transmitted via Communications Means **30** from the Device "A" **10** to other computing devices (e.g. Device "B" **11**, Device "C" **12**, and Device "D" **13**).

The UserB FindMe File **101** is a computer file (e.g. a text file), which is created by the Peer-to-Peer Network Program **40**. The UserB FindMe File **101** contains information relating to the Device "B" **11**, and the user thereof. The UserB FindMe File **101** can contain information such as: the system name of the Device "B" **11**; the name of the user of the Device "B" **11**; the IP address of the Device "B" **11**; etc. The UserB FindMe File **101** is created by the Peer-to-Peer Network Program **40** and can contain information from the User Information File **50**. The UserB FindMe File **101** is created by the Peer-to-Peer Network Program **40** and can contain information obtained by the Peer-to-Peer Network Program **40** through use of the Application Programming Interface **72**. The UserB FindMe File **101** can be created by the Peer-to-Peer Network Program **40** each time the Peer-to-Peer Network Program **40** is started, and any old versions of the UserB FindMe File **101** can be overwritten with a new version, to account for information changes to relating to the Device "B" **11**, and/or the user thereof. The UserB FindMe File **101** can exist in plaintext form and/or ciphertext form. The UserB FindMe File **101** can be encrypted by the Peer-to-Peer Network Program **40** with the UserB Private Key **91** through use of the CryptoAPI **70** of the Device "B" **11**. The UserB FindMe File **101** can be decrypted by the Peer-to-

Peer Network Program **40** with the UserB Public Key **81** through use of the CryptoAPI **70** of the Device "B" **11**. The UserB FindMe File **101** can be decrypted by the Peer-to-Peer Network Program **40** with the UserB Public Key **81** through use of the CryptoAPI **70** of other computing devices (e.g. Device "A" **10**, Device "C" **12**, and Device "D" **13**). The UserB FindMe File **101** can be transmitted via Communications Means **30** from the Device "B" **11** to other computing devices (e.g. Device "A" **10**, Device "C" **12**, and Device "D" **13**).

The UserC FindMe File **102** is a computer file (e.g. a text file), which is created by the Peer-to-Peer Network Program **40**. The UserC FindMe File **102** contains information relating to the Device "C" **12**, and the user thereof. The UserC FindMe File **102** can contain information such as: the system name of the Device "C" **12**; the name of the user of the Device "C" **12**; the IP address of the Device "C" **12**; etc. The UserC FindMe File **102** is created by the Peer-to-Peer Network Program **40** and can contain information from the User Information File **50**. The UserC FindMe File **102** is created by the Peer-to-Peer Network Program **40** and can contain information obtained by the Peer-to-Peer Network Program **40** through use of the Application Programming Interface **72**. The UserC FindMe File **102** can be created by the Peer-to-Peer Network Program **40** each time the Peer-to-Peer Network Program **40** is started, and any old versions of the UserC FindMe File **102** can be overwritten with a new version, to account for information changes to relating to the Device "C" **12**, and/or the user thereof. The UserC FindMe File **102** can exist in plaintext form and/or ciphertext form. The UserC FindMe File **102**

can be encrypted by the Peer-to-Peer Network Program **40** with the UserC Private Key **92** through use of the CryptoAPI **70** of the Device "C" **12**. The UserC FindMe File **102** can be decrypted by the Peer-to-Peer Network Program **40** with the UserC Public Key **82** through use of the CryptoAPI **70** of the Device "C" **12**. The UserC FindMe File **102** can be decrypted by the Peer-to-Peer Network Program **40** with the UserC Public Key **82** through use of the CryptoAPI **70** of other computing devices (e.g. Device "A" **10**, Device "B" **11**, and Device "D" **13**). The UserC FindMe File **102** can be transmitted via Communications Means **30** from the Device "C" **12** to other computing devices (e.g. Device "A" **10**, Device "B" **11**, and Device "D" **13**).

The UserD FindMe File **103** is a computer file (e.g. a text file), which is created by the Peer-to-Peer Network Program **40**. The UserD FindMe File **103** contains information relating to the Device "D" **13**, and the user thereof. The UserD FindMe File **103** can contain information such as: the system name of the Device "D" **13**; the name of the user of the Device "D" **13**; the IP address of the Device "D" **13**; etc. The UserD FindMe File **103** is created by the Peer-to-Peer Network Program **40** and can contain information from the User Information File **50**. The UserD FindMe File **103** is created by the Peer-to-Peer Network Program **40** and can contain information obtained by the Peer-to-Peer Network Program **40** through use of the Application Programming Interface **72**. The UserD FindMe File **103** can be created by the Peer-to-Peer Network Program **40** each time the Peer-to-Peer Network Program **40** is started, and any old versions of the UserD FindMe File **103** can be overwritten with a new version, to

account for information changes to relating to the Device "D" **13**, and/or the user thereof. The UserD FindMe File **103** can exist in plaintext form and/or ciphertext form. The UserD FindMe File **103** can be encrypted by the Peer-to-Peer Network Program **40** with the UserD Private Key **93** through use of the CryptoAPI **70** of the Device "D" **13**. The UserD FindMe File **103** can be decrypted by the Peer-to-Peer Network Program **40** with the UserD Public Key **83** through use of the CryptoAPI **70** of the Device "D" **13**. The UserD FindMe File **103** can be decrypted by the Peer-to-Peer Network Program **40** with the UserD Public Key **83** through use of the CryptoAPI **70** of other computing devices (e.g. Device "A" **10**, Device "B" **11**, and Device "C" **12**). The UserD FindMe File **103** can be transmitted via Communications Means **30** from the Device "D" **13** to other computing devices (e.g. Device "A" **10**, Device "B" **11**, and Device "C" **12**).

The UserA Public Key BLOB **110** is means, which stores public keys (e.g. decryption keys) outside the Cryptographic Service Provider **71**. The UserA Public Key BLOB **110** is also means, which is created by the Peer-to-Peer Network Program **40** using the CryptoAPI **70**. The UserA Public Key BLOB **110** is also means, which can be used to store and transport the UserA Public Key **80**. The UserA Public Key BLOB **110** is also means, which can be transmitted via Communications Means **30** from the Device "A" **10** to other computing devices (e.g. Device "B" **11**, Device "C" **12**, and Device "D" **13**). The Peer-to-Peer Network Program **40**, using the CryptoAPI **70**, of other computing devices (e.g. Device "B" **11**, Device "C" **12**, and Device "D" **13**) can extract the UserA Public Key **80** from the

UserA Public Key BLOB **110**, and save the UserA Public Key **80** to the Cryptographic Service Provider **71** of the other computing devices (e.g. Device "B" **11**, Device "C" **12**, and Device "D" **13**) respectively.

The UserB Public Key BLOB **111** is means, which stores public keys (e.g. decryption keys) outside the Cryptographic Service Provider **71**. The UserB Public Key BLOB **111** is also means, which is created by the Peer-to-Peer Network Program **40** using the CryptoAPI **70**. The UserB Public Key BLOB **111** is also means, which can be used to store and transport the UserB Public Key **81**. The UserB Public Key BLOB **111** is also means, which can be transmitted via Communications Means **30** from the Device "B" **11** to other computing devices (e.g. Device "A" **10**, Device "C" **12**, and Device "D" **13**). The Peer-to-Peer Network Program **40**, using the CryptoAPI **70**, of other computing devices (e.g. Device "A" **10**, Device "C" **12**, and Device "D" **13**) can extract the UserB Public Key **81** from the UserB Public Key BLOB **111**, and save the UserB Public Key **81** to the Cryptographic Service Provider **71** of the other computing devices (e.g. Device "A" **10**, Device "C" **12**, and Device "D" **13**) respectively.

The UserC Public Key BLOB **112** is means, which stores public keys (e.g. decryption keys) outside the Cryptographic Service Provider **71**. The UserC Public Key BLOB **112** is also means, which is created by the Peer-to-Peer Network Program **40** using the CryptoAPI **70**. The UserC Public Key BLOB **112** is also means, which can be used to store and transport the UserC Public Key **82**. The

UserC Public Key BLOB **112** is also means, which can be transmitted via Communications Means **30** from the Device "C" **12** to other computing devices (e.g. Device "A" **10**, Device "B" **11**, and Device "D" **13**). The Peer-to-Peer Network Program **40**, using the CryptoAPI **70**, of other computing devices (e.g. Device "A" **10**, Device "B" **11**, and Device "D" **13**) can extract the UserC Public Key **82** from the UserC Public Key BLOB **112**, and save the UserC Public Key **82** to the Cryptographic Service Provider **71** of the other computing devices (e.g. Device "A" **10**, Device "B" **11**, and Device "D" **13**) respectively.

The UserD Public Key BLOB **113** is means, which stores public keys (e.g. decryption keys) outside the Cryptographic Service Provider **71**. The UserD Public Key BLOB **113** is also means, which is created by the Peer-to-Peer Network Program **40** using the CryptoAPI **70**. The UserD Public Key BLOB **113** is also means, which can be used to store and transport the UserD Public Key **83**. The UserD Public Key BLOB **113** is also means, which can be transmitted via Communications Means **30** from the Device "D" **13** to other computing devices (e.g. Device "A" **10**, Device "B" **11**, and Device "C" **12**). The Peer-to-Peer Network Program **40**, using the CryptoAPI **70**, of other computing devices (e.g. Device "A" **10**, Device "B" **11**, and Device "C" **12**) can extract the UserD Public Key **83** from the UserD Public Key BLOB **113**, and save the UserD Public Key **83** to the Cryptographic Service Provider **71** of the other computing devices (e.g. Device "A" **10**, Device "B" **11**, and Device "C" **12**) respectively.

The File Information Stream **120** is a stream of informational data, which is embedded into one of multiple streams, which compose the entire file structure of the computer file, which the File Information Stream **120** is associated, and is stored on Storage **23** (e.g. Microsoft NTFS storage device). The File Information Stream **120** can also be a stream of data, which is appended to, associated with, or related to the computer file. The File Information Stream **120** can include information such as: name of computer file; date created; author; system on which the computer file was created; description of the computer file; etc. The File Information Stream **120** is created by the Peer-to-Peer Network Program **40** through use of the Application Programming Interface **72**. The File Information Stream **120** can be accessed by the Peer-to-Peer Network Program **40** through use of the Application Programming Interface **72**, and the related information displayed to the user via the User Interface **20**. The data can be a digital signal of any type of data (business, technical, pleasure), for instance, including but not limited to, a video digital signal, a audio digital signal.

The E-mail Program **130** is means, which enables the transfer or exchange of computer messages from one computing device to another computing device, utilizing certain Transport Control Protocol/Internet Protocol protocols (e.g. Simple Mail Transfer Protocol, Post Office Protocol 3, Internet Message Access Protocol, etc.). Computer messages transmitted via the E-mail Program **130** is can include text information, attached computer files, etc. The E-mail Program **130** is also means, which enables a user of a computing device (e.g. Device "A" **10**, Device "B" **11**, Device "C" **12**, and

Device "D" **13**) to manually and electronically transfer a public key BLOB (e.g. the UserA Public Key BLOB **110**, UserB Public Key BLOB **111**, UserC Public Key BLOB **112**, and UserD Public Key BLOB **113**) from the originating computing device to another computing device. The E-mail Program **130** is also means, which enables a user of a computing device (e.g. Device "A" **10**, Device "B" **11**, Device "C" **12**, and Device "D" **13**) to manually and electronically transfer of the name of the user FindMe file (e.g. the UserA FindMe File **100**, UserB FindMe File **101**, UserC FindMe File **102**, and UserD FindMe File **103**) from the originating computing device to another computing device. The E-mail Program **130** is also means, which enables the Peer-to-Peer Network Program **40** of a computing device (e.g. Device "A" **10**, Device "B" **11**, Device "C" **12**, and Device "D" **13**) to automatically and electronically transfer a public key BLOB (e.g. the UserA Public Key BLOB **110**, UserB Public Key BLOB **111**, UserC Public Key BLOB **112**, and UserD Public Key BLOB **113**) from the originating computing device to another computing device. The E-mail Program **130** is also means, which enables the Peer-to-Peer Network Program **40** of a computing device (e.g. Device "A" **10**, Device "B" **11**, Device "C" **12**, and Device "D" **13**) to automatically and electronically transfer of the name of the user FindMe file (e.g. the UserA FindMe File **100**, UserB FindMe File **101**, UserC FindMe File **102**, and UserD FindMe File **103**) from the originating computing device to another computing device.

The Trusted Member List **140** is a listing within the Peer-to-Peer Network Program **40**, which associates the user FindMe files (e.g. the UserA FindMe File **100**, UserB FindMe File **101**, UserC

FindMe File **102**, and UserD FindMe File **103**) with the IP address of the computing device (e.g. Device "A" **10**, Device "B" **11**, Device "C" **12**, and Device "D" **13**) respectively. The Trusted Member List **140** can be updated each time the Peer-to-Peer Network Program **40** is started. The Trusted Member List **140** can be updated, or refreshed, on demand by the user of the Peer-to-Peer Network Program **40**. The IP addresses listed in the Trusted Member List **140** are utilized by the Peer-to-Peer Network Program **40** to establish a trusted and decentralized peer-to-peer network.

The FTP Protocol **150** is a standard Internet protocol, known as the File Transfer Protocol. The FTP Protocol **150** is generally known as the simplest way to transmit and/or exchange computer files between computing devices on the Internet. The Peer-to-Peer Network Program **40** can utilize the FTP Protocol **150** to transmit and/or exchange computer files via Communications Means **30**, between computing devices (e.g. Device "A" **10**, Device "B" **11**, Device "C" **12**, and Device "D" **13**). The Peer-to-Peer Network Program **40** can utilize the FTP Protocol **150** to communicate, via Communications Means **30**, between computing devices (e.g. Device "A" **10**, Device "B" **11**, Device "C" **12**, and Device "D" **13**).

The HTTP Protocol **151** is a standard Internet protocol, known as the Hypertext Transfer Protocol. The HTTP Protocol **151** is a common way to transmit and/or exchange computer files between computing devices on the Internet. The Peer-to-Peer Network Program **40** can utilize the HTTP Protocol **151** to transmit and/or exchange computer files via Communications Means **30**, between

computing devices (e.g. Device "A" **10**, Device "B" **11**, Device "C" **12**, and Device "D" **13**). The Peer-to-Peer Network Program **40** can utilize the HTTP Protocol **151** to communicate, via Communications Means **30**, between computing devices (e.g. Device "A" **10**, Device "B" **11**, Device "C" **12**, and Device "D" **13**).

The Chat Protocol **152** is a system for electronic communications between computing devices that involves a set of rules and conventions and is known as Internet Relay Chat (IRC) and is a standard Internet protocol. The Chat Protocol **152** is generally used for the real-time exchange of typed-in messages between a user of a computing device (e.g. Device "A" **10**) and other users of other computing devices (e.g. Device "B" **11**, Device "C" **12**, and/or Device "D" **13**), utilizing a communications means (e.g. the Communications Means **30**). The Chat Protocol **152** can be used by the Peer-to-Peer Network Program **40** to establish a private chat session between a user of a computing device (e.g. Device "A" **10**) and other users of other computing devices (e.g. Device "B" **11**, Device "C" **12**, and/or Device "D" **13**), in conjunction with the trusted and decentralized peer-to-peer network the Peer-to-Peer Network Program **40** is able to establish.

The File Sharing Protocols **152** are set of rules and conventions used to leverage other Internet protocols (e.g. the FTP Protocol **150**, HTTP Protocol **151**, etc.) to search designated file folders on a storage device (e.g. the Storage **23**) of a computing device (e.g. Device "A" **10**, Device "B" **11**, Device "C" **12**, and Device "D" **13**) for computer files which match a query inputted by

a user of a computing device. The File Sharing Protocols **152** can be used by the Peer-to-Peer Network Program **40** to execute a search and retrieval of a computer file based on requests manually inputted by a user of a computing device (e.g. Device "A" **10**), of designated file folders (e.g. Trusted Search Folder **62**) on a storage device (e.g. the Storage **23**) of other computing devices (e.g. Device "B" **11**, Device "C" **12**, and/or Device "D" **13**), in conjunction with the trusted and decentralized peer-to-peer network the Peer-to-Peer Network Program **40** is able to establish.

By means of the User Interface **20**, the user of a computing device with a connection to the Setup.exe File **41** (e.g. the Setup.exe File **41** being on CD-ROM, the Storage **23**, etc.), double clicks on the Setup.exe File **41** to start the installation process. Next the Setup.exe File **41** commences to copy the Peer-to-Peer Network Program **40** onto the Storage **23**. Next the Setup.exe File **41** requests information from the user and the user inputs that information via the User Interface **20**. Next the Setup.exe File **41** saves that user information as a User Information File **50** on the Storage **23**. Next the Setup.exe File **41** saves makes any necessary modifications to the system registry of the Operating System **25**. Next the Setup.exe File **41** creates the FindMe Folder **60**, the Searched FindMe Results Folder **61**, and the Trusted Search Folder **62** on the Storage **23**. These steps are performed for each computing device (e.g. Device "A" **10**, Device "B" **11**, Device "C" **12**, Device "D" **13**) intended to run the Peer-to-Peer Network Program **40**.

Next, and upon first use of the Peer-to-Peer Network Program **40**, the Peer-to-Peer Network Program **40** prompts the user of the host computing device to input a name which identifies that user (e.g. UserA). Next, the Peer-to-Peer Network Program **40** calls cryptographic functionality of the CryptoAPI **70** (e.g. the Peer-to-Peer Network Program **40** calls the "CryptAcquireContext" function of the Windows CryptoAPI) of the Operating System **25** to create a public/private key pair. Next, the CryptoAPI **70** returns a handle to the Cryptographic Service Provider **71** (e.g. the Windows CryptoAPI returns a handle to the Microsoft RSA/Schannel Cryptographic Service Provider to the Peer-to-Peer Network Program **40**). Next, the Peer-to-Peer Network Program **40** calls cryptographic functionality of the CryptoAPI **70** of the Operating System **25** to instruct the Cryptographic Service Provider **71** to execute the creation of the public/private key pair (e.g. the Peer-to-Peer Network Program **40** calls the "CryptGenKey" function of the Windows CryptoAPI). Next, the CryptoAPI **70** instructs the Cryptographic Service Provider **71** (e.g. the Windows CryptoAPI instructs the Microsoft RSA/Schannel Cryptographic Service Provider to create the public/private key pair (e.g. UserAPublic Key **80** and UserA Private Key **90**)).

Next, the Peer-to-Peer Network Program **40** creates a plaintext version of the UserA FindMe File **100** (e.g. a text file), and at this point UserA FindMe File **100** is a blank file. Next, the Peer-to-Peer Network Program **40** calls functionality of the Application Programming Interface **72** of the Operating System **25** and obtains identifiable network information on the host computing

device (e.g. Device "A" **10**) (e.g. the Peer-to-Peer Network Program **40** calls the "CurrentIpAddress" function of the Windows API). Next, the Application Programming Interface **72** queries the Operating System **25** and obtains identifiable network information on the host computing device (e.g. Device "A" **10**), then the Application Programming Interface **72** transmits the identifiable network information to the Peer-to-Peer Network Program **40**. Next, the Peer-to-Peer Network Program **40** writes the identifiable network information into the plaintext version of the UserA FindMe File **100**. Next, the Peer-to-Peer Network Program **40** calls cryptographic functionality of the CryptoAPI **70** (e.g. the Peer-to-Peer Network Program **40** calls the "CPEncrypt" function of the Windows CryptoAPI) of the Operating System **25** to encrypt the UserA FindMe File **100** with a private key (e.g. the UserA Private Key **90**). At this point the Peer-to-Peer Network Program **40** is configured to execute other embodiments of the invention.

Next, and each time, the user initiates the Peer-to-Peer Network Program **40** by means of the User Interface **20**, the Peer-to-Peer Network Program **40** automatically searches each of the Trusted Search Folders **62** for computer files lacking a File Info Stream **120** through use of functionality of the Application Programming Interface **72** of the Operating System **25** (e.g. the Peer-to-Peer Network Program **40** calls the "FileSystemObject" object model of the Windows API to open, write, create, and test computer files). Next, and for each computer file in each of the Trusted Search Folders **62** that lacks a File Info Stream **120**, the Peer-to-Peer Network Program **40** calls functionality of the Application

Programming Interface **72** of the Operating System **25**, then opens the User Information File **50**, then writes information from the User Information File **50** into the File Info Stream **120** (e.g. the Peer-to-Peer Network Program **40** calls the "FileSystemObject" object model of the Windows API to open, write, create, and test computer files). The Peer-to-Peer Network Program **40** also enables the user of the host computing device to manually input other information into the File Info Stream **120** for each related computer file and accomplishes this task by calling functionality of the Application Programming Interface **72** of the Operating System **25**, then writes the information the user manually inputted into the File Info Stream **120** (e.g. the Peer-to-Peer Network Program **40** calls the "FileSystemObject" object model of the Windows API to open, write, create, and test computer files).

Next, the user (e.g. UserA) of a computing device (e.g. Device "A" **10**) inputs a command to the Peer-to-Peer Network Program **40** via the User Interface **20**, to export their related public key (e.g. UserA Public Key **80**) to a recipient or recipients, being a user of a another computing device or to other users of other computing devices and the UserA inputs the E-mail addresses of the recipient or recipients. Next, the Peer-to-Peer Network Program **40** calls functionality of the CryptoAPI **70** of the Operating System **25** to create a computer file capable of transporting the UserA Public Key **80** (e.g. the Peer-to-Peer Network Program **40** calls the "CryptExportKey" function of the Windows CryptoAPI to generate a public key BLOB of UserA Public Key **110**). Next, the Peer-to-Peer Network Program **40** instructs the E-mail Program **130** of the Device "A" **10** to transmit to the recipient or recipients the name of the

UserA FindMe File **110** and a copy of the UserA Public Key BLOB **110**. Next, the E-mail Program **130** of a recipient's computing device (e.g. Device "B" **11**) receives from transmission from the UserA the name of the UserA FindMe File **110** and a copy of the UserA Public Key BLOB **110**.

Next, the user of the Device "B" **11** (upon receipt of E-mail transmission from the UserA the name of the UserA FindMe File **110** and the copy of the UserA Public Key BLOB **110**) instructs the Peer-to-Peer Network Program **40** via the User Interface **20**, to import the UserA Public Key **80**. Next, the Peer-to-Peer Network Program **40** instructs the CryptoAPI **70** of the Operating System **25** of the Device "B" **11** to import the UserA Public Key **80** from the UserA Public Key BLOB **110** to the Cryptographic Service Provider **71** (e.g. the Peer-to-Peer Network Program **40** calls the "CryptImportKey" function of the Windows CryptoAPI, then the Windows CryptoAPI extracts the UserA Public Key **80** from the UserA Public Key BLOB **110**, then the Windows CryptoAPI imports and the UserA Public Key **80** to the Microsoft RSA/Schannel Cryptographic Service Provider). Next the Peer-to-Peer Network Program **40** writes an entry in the Trusted Member List **140** establishing (or maping) a relationship between the name of the UserA FindMe File **100** (as received from E-mail transmission) and the UserA Public Key **80**.

Next, the user of the Device "A" **10** (e.g. UserA) instructs the Peer-to-Peer Network Program **40** via the User Interface **20** of Device "A" **10**, to search (via the Communications Means **30**) for members (who are then connected to the Communications

Means **30**) listed in the Trusted Member List **140** (being the members of the trusted peer-to-peer network the Peer-to-Peer Network Program **40** is capable of establishing) via connected to the Communications Means **30**. Next, the Peer-to-Peer Network Program **40** accesses the list of computer file names in the Trusted Member List **140** and executes a search request of an un-trusted peer-to-peer network for computer files matching the name of the computer files listed in the Trusted Member List **140** utilizing the File Sharing Protocols **153**. Next, the Peer-to-Peer Network Program **40** of another computing device (e.g. Device "B" **11**) receives the search request for various computer files (e.g. user "FindMe" files) and transmits the computer files matching the search request (e.g. UserB FindMe File **101**) to the Device "A" **10** utilizing transmission protocols (e.g. FTP Protocol **150**, HTTP Protocol **151**).

Next, the Peer-to-Peer Network Program **40** of the Device "A" **10** receives from transmission the UserB FindMe File **101**, then saves the UserB FindMe File **101** in the FindMe Results Folder **61** in Storage **23** of the Device "A" **10**. Next, the Peer-to-Peer Network Program **40** instructs the CryptoAPI **70** of the Operating System **25** of the Device "A" **10** to decrypt the UserA FindMe File **101** using the public key as associated with the UserA FindMe File **101** in the Trusted Member List **140** through use of the Cryptographic Service Provider **71** (e.g. the Peer-to-Peer Network Program **40** calls the "CPDecrypt" function of the Windows CryptoAPI, then the Windows CryptoAPI calls the UserB Public Key **81** from the Microsoft RSA/Schannel Cryptographic Service Provider, then the Windows CryptoAPI decrypts the UserB FindMe File **101** creating a plaintext

version of the UserB FindMe File **101**). Next, the Peer-to-Peer Network Program **40** of the Device "A" **10** opens the plaintext version of the UserB FindMe File **101**) and reads the identifiable network information (e.g. the IP address of Device "B" **11**) then writes an entry in the Trusted Member List **140** containing: the identifiable network information of Device "B" **11**, the name of the UserB FindMe File **101**, and name of the UserB. At this point, the user of the Device "A" **10** is able to use the Peer-to-Peer Network Program **40** to establish a trusted and decentralized peer-to-peer network, utilizing the IP addresses listed on a member-by-member basis in the Trusted Member List **140**.

Users of the Peer-to-Peer Network Program **40** can utilize the trusted and decentralized peer-to-peer network to establish trusted chat sessions. This is accomplished when the user of a computing device (e.g. Device "A" **10**) instructs the Peer-to-Peer Network Program **40** via the User Interface **20**, to establish a chat session with user defined members listed in the Trusted Member List **140**. Next, the Peer-to-Peer Network Program **40** reads the identifiable network information (e.g. the IP address of the members) in the Trusted Member List **140** of the members, then, using the Chat Protocol **152**, the Peer-to-Peer Network Program **40** transmits a chat session request to the computing devices (e.g. Device "B" **11**, Device "C" **12**, Device "D" **13**, etc.) of the members (e.g. UserB, UserC, UserD, etc.). Each computing device that is then: connected to the Communications Means **30** utilizing identifiable network information (e.g. IP address) matching the identifiable network information listed on a member-by-member basis in the Trusted Member List **140** of the Device "A" **10**; running the

Peer-to-Peer Network Program **40**; and receives the transmitted chat session request from Device "A" **10** utilizing the Chat Protocol **152**; then notifies the user (e.g. UserB, UserC, UserD, etc.) of that particular computing device (e.g. Device "B" **11**, Device "C" **12**, Device "D" **13**, etc.) via the User Interface **20** of their respective computing device, that a chat session has been initiated by a member of the trusted network. Next, the user (e.g. UserB, UserC, UserD, etc.) of that particular computing device (e.g. Device "B" **11**, Device "C" **12**, Device "D" **13**, etc.) instructs the Peer-to-Peer Network Program **40** to either accept or deny the chat session, via the User Interface **20** of their respective computing device. If the user (e.g. UserB, UserC, UserD, etc.) of that particular computing device (e.g. Device "B" **11**, Device "C" **12**, Device "D" **13**, etc.) instructs the Peer-to-Peer Network Program **40** to deny the chat session, via the User Interface **20** of their respective computing device, then the Peer-to-Peer Network Program **40** denies the chat session utilizing the Chat Protocol **152** and transmits a denial notification to the Peer-to-Peer Network Program **40** of the Device "A" **10**, then the Peer-to-Peer Network Program **40** of the Device "A" **10** notifies the user (e.g. UserA) of the Device "A" **10** that the chat session has been denied, then the Peer-to-Peer Network Program **40** of the Device "B" **11** ends the chat session utilizing the Chat Protocol **152**. If the user (e.g. UserB, UserC, UserD, etc.) of that particular computing device (e.g. Device "B" **11**, Device "C" **12**, Device "D" **13**, etc.) instructs the Peer-to-Peer Network Program **40** to accept the chat session, via the User Interface **20** of their respective computing device, then the Peer-to-Peer Network Program **40** accepts the chat session utilizing the Chat Protocol **152**, then

the Peer-to-Peer Network Program **40** maintains the chat session utilizing the Chat Protocol **152**.

Users of the Peer-to-Peer Network Program **40** can utilize the trusted and decentralized peer-to-peer network to conduct trusted file sharing or searching. This is accomplished when the user of a computing device (e.g. Device "A" **10**) instructs the Peer-to-Peer Network Program **40** via the User Interface **20**, to establish a file search of the computing devices controlled or operated by the members listed in the Trusted Member List **140** by inputting the name of the computer file desired via the User Interface **20**. Next, the Peer-to-Peer Network Program **40** reads the identifiable network information (e.g. the IP address of the members) in the Trusted Member List **140** of the members, then, using the File Sharing Protocols **153**, the Peer-to-Peer Network Program **40** transmits a search request, for the computer file requested by UserA, to the computing devices listed in the Trusted Member List **140** (e.g. Device "B" **11**, Device "C" **12**, Device "D" **13**, etc.) of the members (e.g. UserB, UserC, UserD, etc.). Each computing device that is then: connected to the Communications Means **30** utilizing identifiable network information (e.g. IP address) matching the identifiable network information listed on a member-by-member basis in the Trusted Member List **140** of the Device "A" **10**; running the Peer-to-Peer Network Program **40**; and receives the transmitted file search request from Device "A" **10** utilizing the File Sharing Protocols **153**; then the Peer-to-Peer Network Program **40** of that particular computing device (e.g. Device "B" **11**, Device "C" **12**, Device "D" **13**, etc.) searches for the requested file in the Trusted Search Folder **62**.

If the requested computer file is not located by the Peer-to-Peer Network Program **40** of a particular computing device (e.g. Device "B" **11**, Device "C" **12**, Device "D" **13**, etc.) in the respective Trusted Search Folder **62**, then the Peer-to-Peer Network Program **40** does not reply. If the requested computer file is located by the Peer-to-Peer Network Program **40** of a particular computing device (e.g. Device "B" **11**, Device "C" **12**, Device "D" **13**, etc.) in the respective Trusted Search Folder **62**, then the Peer-to-Peer Network Program **40** transmits the name of the matching computer file along with any information found in the File Info Stream **120** related to the matching computer file, to the Peer-to-Peer Network Program **40** of the Device "A" **10**. Next, the Peer-to-Peer Network Program **40** of the Device "A" **10** displays via the User Interface **20**, the computer file names, and any information found in the File Info Stream **120** related to the matching computer file, received from any and/or all computing devices (e.g. Device "B" **11**, Device "C" **12**, Device "D" **13**, etc.) that have responded to the search request. Next, the user of the Device "A" **10** selects, via the User Interface **20**, which computer file or computer files the Peer-to-Peer Network Program **40** is to download via the trusted peer-to-peer network. Next, the Peer-to-Peer Network Program **40** of the Device "A" **10** transmits a download request using communication protocols (e.g. FTP Protocol **150**, HTTP Protocol **151**) to the computing device and/or computing devices, by means of the identifiable network information (e.g. the IP address of the computing device and/or computing devices) in the Trusted Member List **140** and as associated with the specific computer file or computer files selected by the UserA. Next, the Peer-to-Peer Network Program **40** of the computing device

and/or computing devices respectively (e.g. Device "B" **11**, Device "C" **12**, Device "D" **13**, etc.), containing the computer file or computer files selected by the UserA, initiates a download of the computer file or computer files using communication protocols (e.g. FTP Protocol **150**, HTTP Protocol **151**). Next, the Peer-to-Peer Network Program **40** of the Device "A" **10** receives and saves the download computer file or computer files to the Trusted Search Folder **62**.

Although the invention has been described in detail in the foregoing embodiments for the purpose of illustration, it is to be understood that such detail is solely for that purpose and that variations can be made therein by those skilled in the art without departing from the spirit and scope of the invention except as it may be described by the following claims.

WHAT IS CLAIMED IS:

1. A system to establish a trusted and decentralized peer-to peer network comprising:

communication means;

n user computing devices connected to the communication means, where n is greater than or equal to 1 and is an integer; and

a host computing device connected to the communication means having a mechanism to establish a decentralized trusted communications network with at least 2 of the n user computing devices through which digital signals are shared securely between the host computing device and the 2 user computing devices of the trusted communications network.

2. A method for establishing a trusted and decentralized peer-to-peer network comprising the steps of:

sending a public key from a host computing device to communication means connected to the host computing device;

receiving the public key at a first user computing device connected to the communication means; and

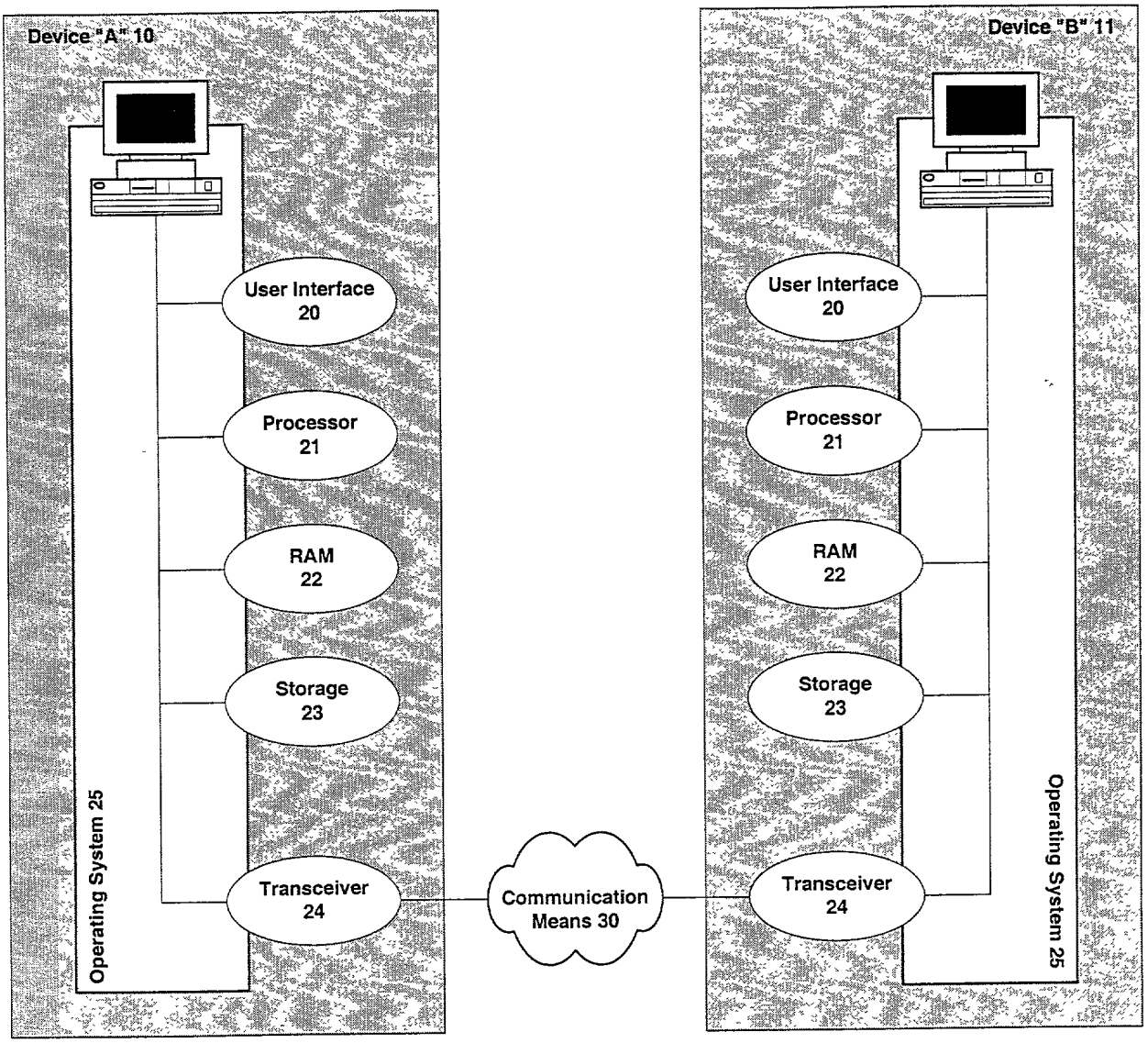
receiving the public key at a second user computing device to establish a decentralized trusted communications network between the host computing device, the first and the second computing device through which digital signals are shared securely between the host computing device, the first user computing device and the second user computing device.

ABSTRACT OF THE DISCLOSURE

METHOD AND SYSTEM FOR ESTABLISHING A TRUSTED AND DECENTRALIZED PEER-TO-PEER NETWORK

The present invention offers a new and improved method and system to establish a trusted and decentralized peer-to-peer network for: the sharing of computer files between and among computing devices; trusted chat sessions; and for other applications of trusted peer-to-peer networks.

Fig. 1


Device "A" 10
User Interface 20
Processor 21
RAM 22
Storage 23
Operating System 25
Transceiver 24
Communication Means 30
Device "B" 11
User Interface 20
Processor 21
RAM 22
Storage 23
Operating System 25
Transceiver 24

Initial Install of Software

Fig. 2

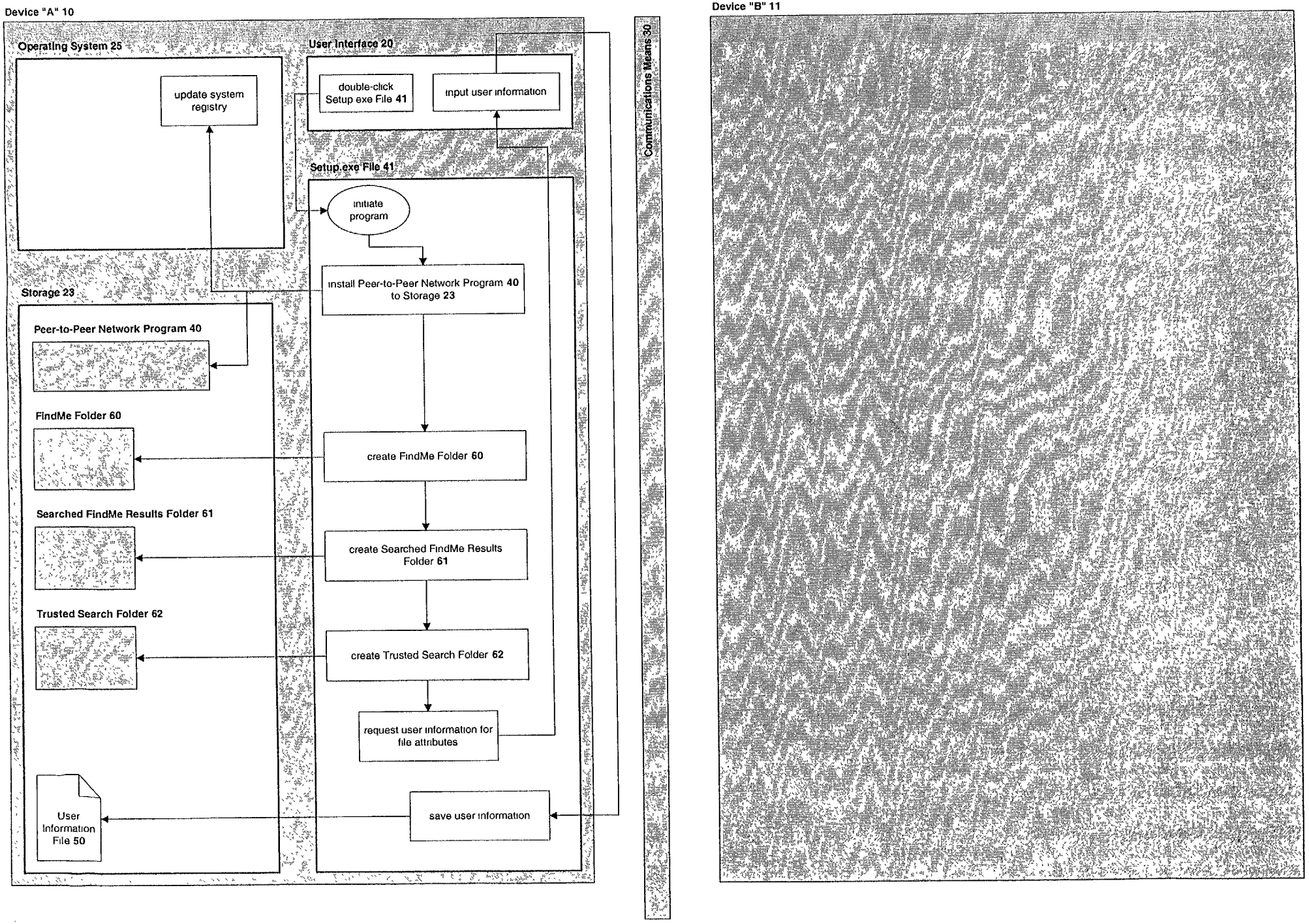


Fig. 3

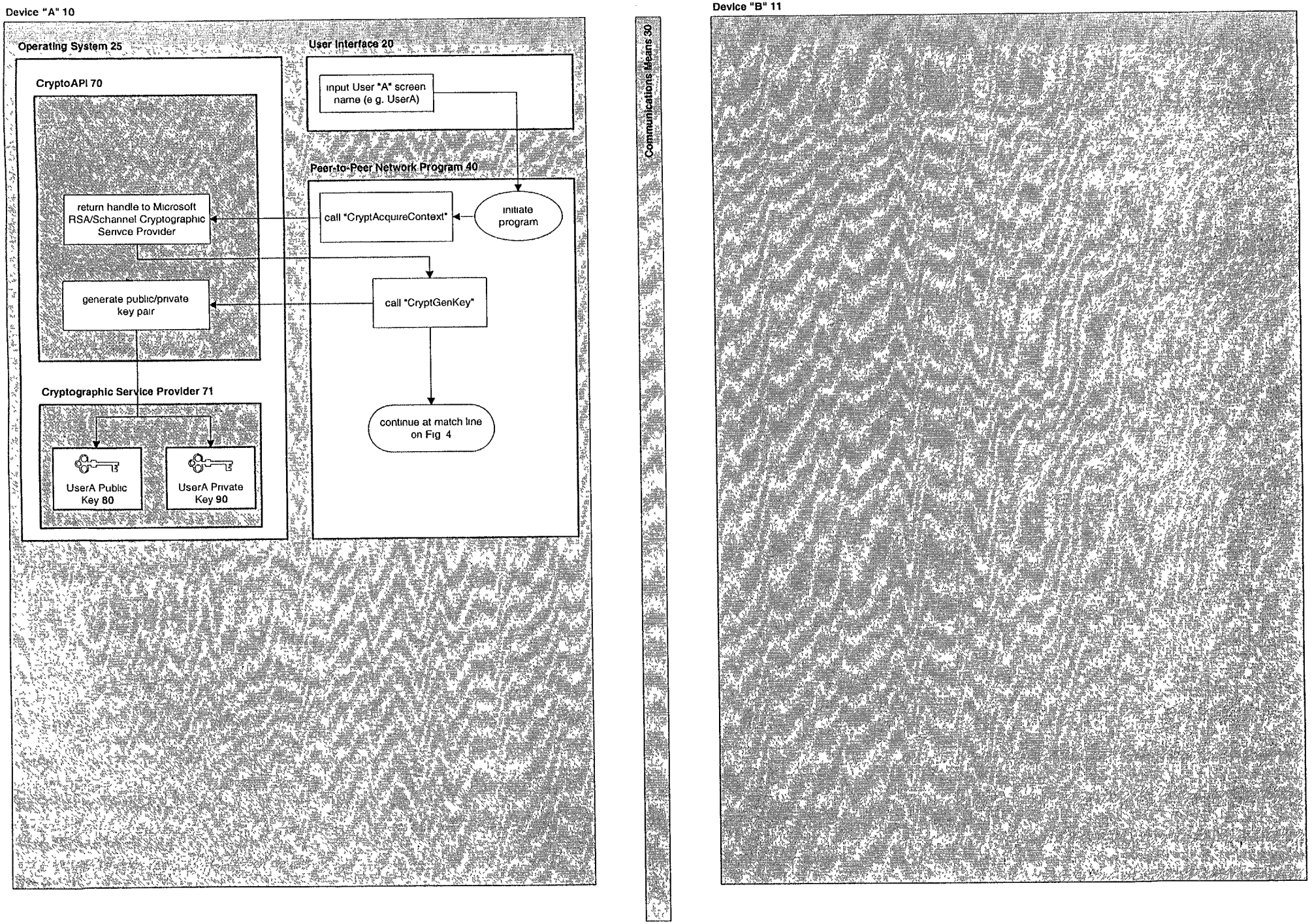

Device "A" 10
Operating System 25
User Interface 20
input User "A" screen name (e.g. UserA)
CryptoAPI 70
Peer-to-Peer Network Program 40
return handle to Microsoft RSA/Schannel Cryptographic Service Provider
call "CryptAcquireContext"
initiate program
generate public/private key pair
call "CryptGenKey"
Cryptographic Service Provider 71
continue at match line on Fig 4
UserA Public Key 80
UserA Private Key 90
Communications Means 30
Device "B" 11

Fig. 4

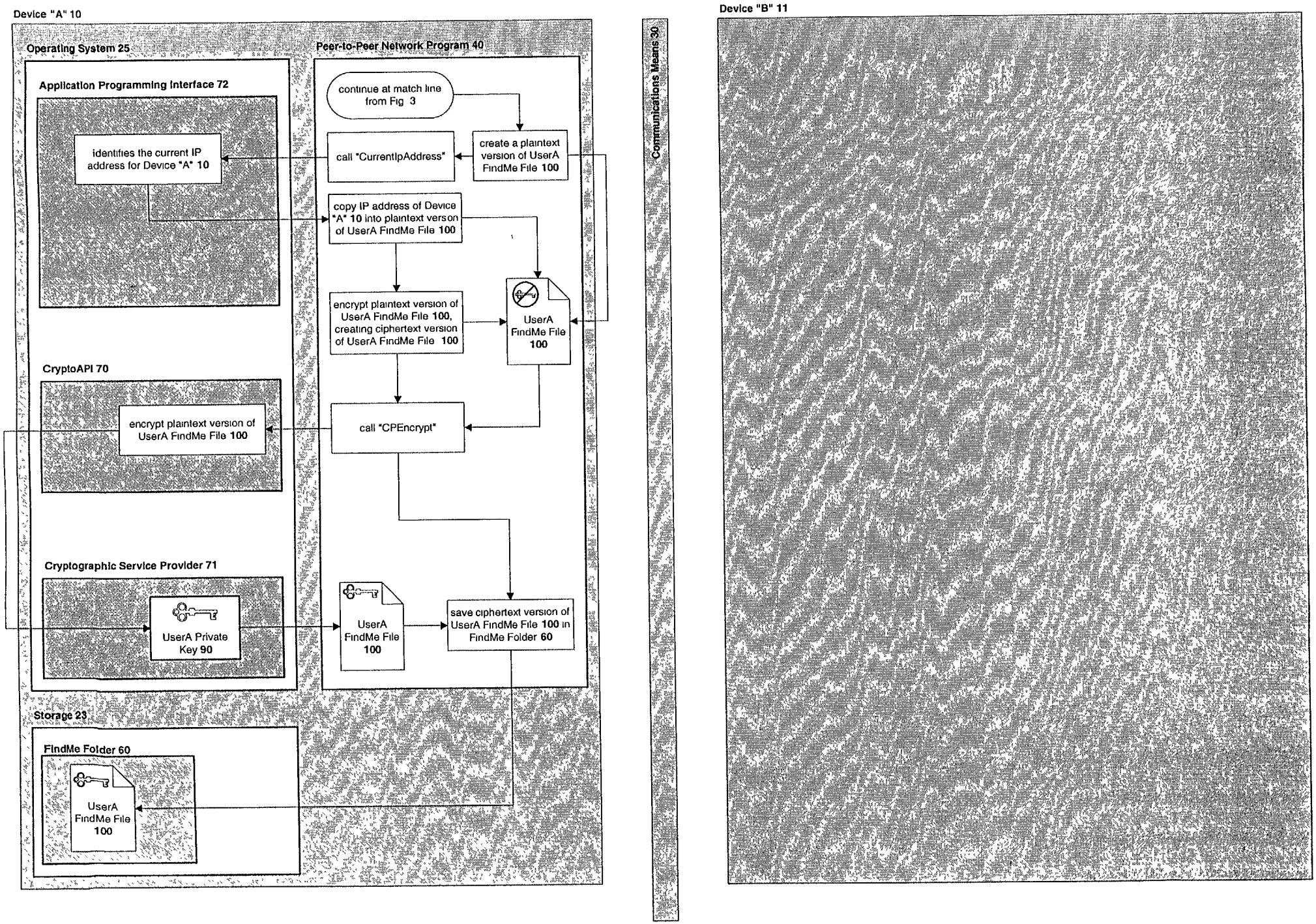
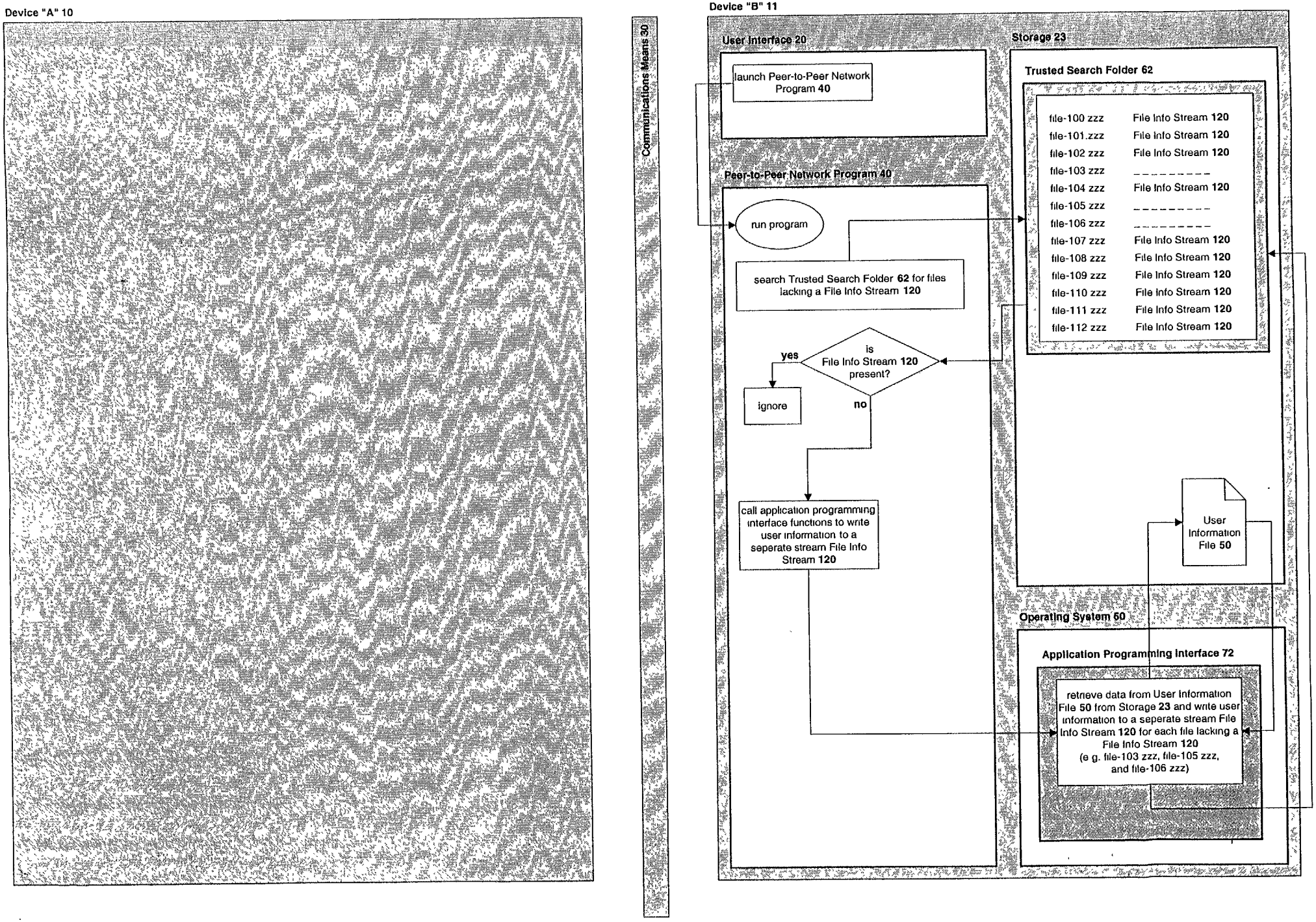

Device "A" 10
Operating System 25
Application Programming Interface 72
identifies the current IP address for Device "A" 10
CryptoAPI 70
encrypt plaintext version of UserA FindMe File 100
Cryptographic Service Provider 71
UserA Private Key 90
Storage 23
FindMe Folder 60
UserA FindMe File 100
Peer-to-Peer Network Program 40
continue at match line from Fig 3
call "CurrentIpAddress"
create a plaintext version of UserA FindMe File 100
copy IP address of Device "A" 10 into plaintext verson of UserA FindMe File 100
encrypt plaintext version of UserA FindMe File 100, creating ciphertext version of UserA FindMe File 100
UserA FindMe File 100
call "CPEncrypt"
UserA FindMe File 100
save ciphertext version of UserA FindMe File 100 in FindMe Folder 60
Communications Means 30
Device "B" 11

Fig. 5


Device "A" 10
Device "B" 11
Communications Means 30
User Interface 20
launch Peer-to-Peer Network Program 40
Storage 23
Trusted Search Folder 62
file-100 zzz File Info Stream 120
file-101.zzz File Info Stream 120
file-102 zzz File Info Stream 120
file-103 zzz
file-104 zzz File Info Stream 120
file-105 zzz
file-106 zzz
file-107 zzz File Info Stream 120
file-108 zzz File Info Stream 120
file-109 zzz File Info Stream 120
file-110 zzz File Info Stream 120
file-111 zzz File Info Stream 120
file-112 zzz File Info Stream 120
Peer-to-Peer Network Program 40
run program
search Trusted Search Folder 62 for files lacking a File Info Stream 120
is File Info Stream 120 present?
yes
ignore
no
call application programming interface functions to write user information to a seperate stream File Info Stream 120
User Information File 50
Operating System 60
Application Programming Interface 72
retrieve data from User Information File 50 from Storage 23 and write user information to a seperate stream File Info Stream 120 for each file lacking a File Info Stream 120 (e g. file-103 zzz, file-105 zzz, and file-106 zzz)

Distribute Public Keys

Fig. 6

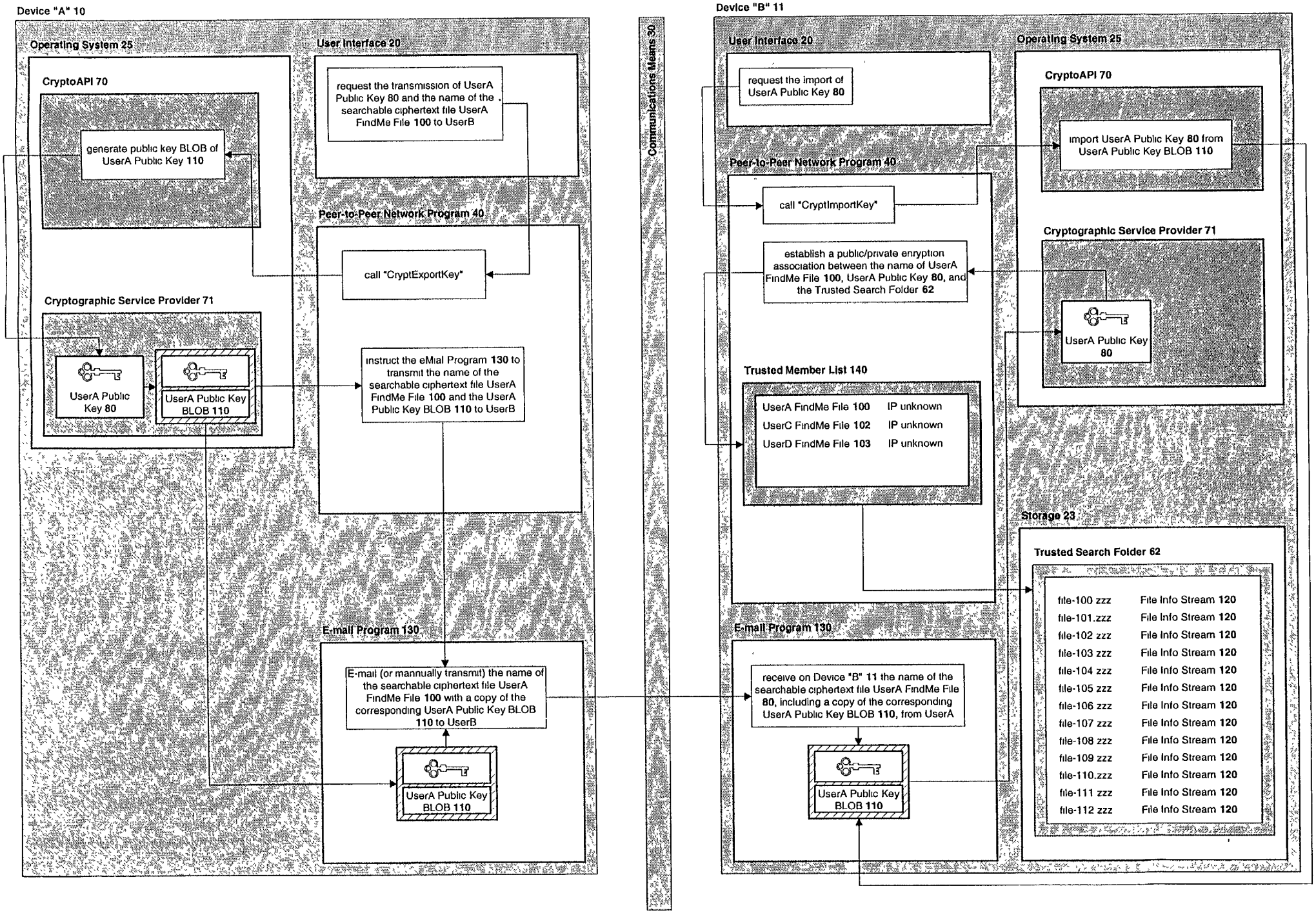


Fig. 7

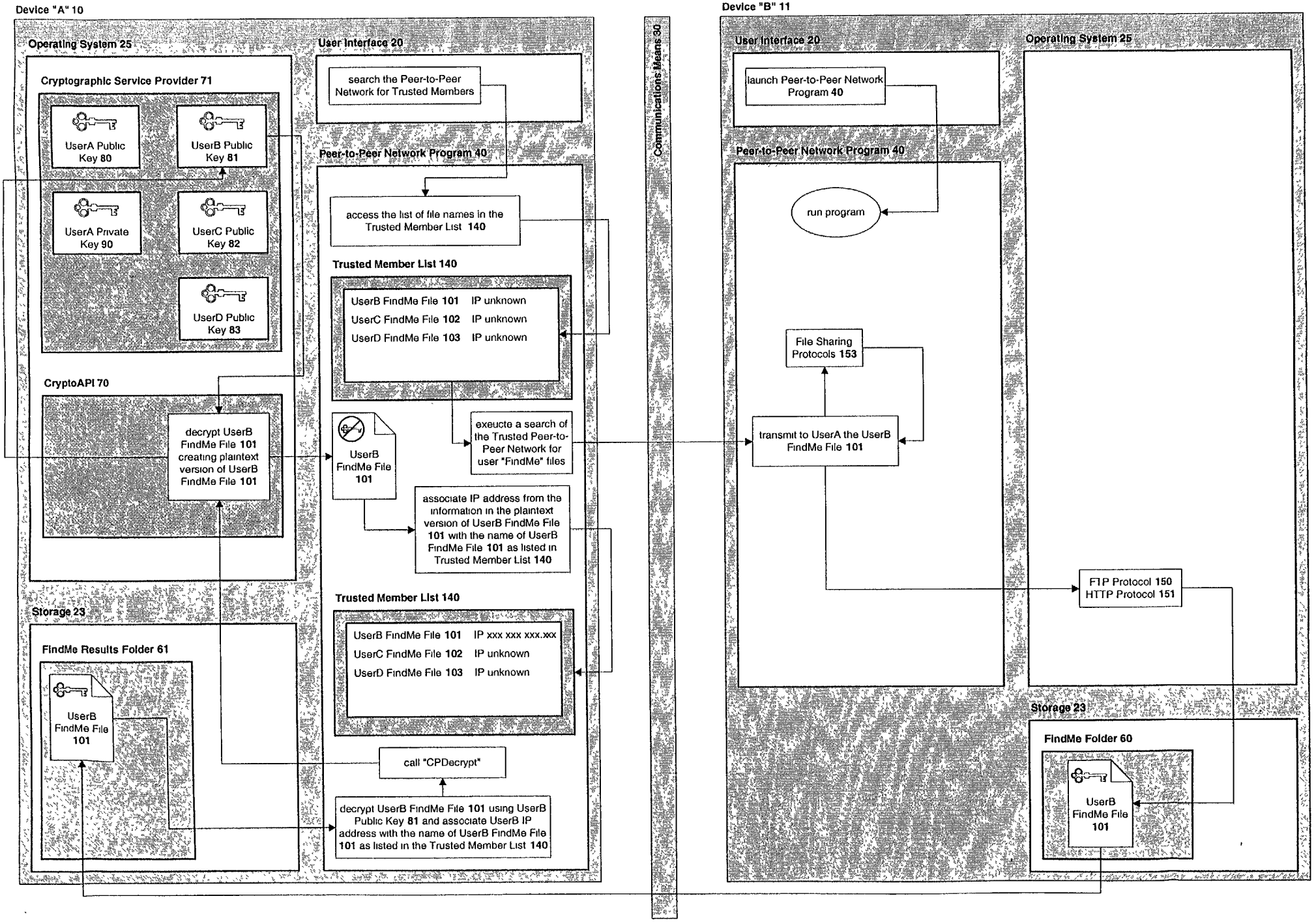

Device "A" 10
Operating System 25
Cryptographic Service Provider 71
UserA Public Key 80
UserB Public Key 81
UserA Private Key 90
UserC Public Key 82
UserD Public Key 83
CryptoAPI 70
decrypt UserB FindMe File 101 creating plaintext version of UserB FindMe File 101
Storage 23
FindMe Results Folder 61
UserB FindMe File 101
User Interface 20
search the Peer-to-Peer Network for Trusted Members
Peer-to-Peer Network Program 40
access the list of file names in the Trusted Member List 140
Trusted Member List 140
UserB FindMe File 101 IP unknown
UserC FindMe File 102 IP unknown
UserD FindMe File 103 IP unknown
UserB FindMe File 101
exeucte a search of the Trusted Peer-to-Peer Network for user "FindMe" files
associate IP address from the information in the plaintext version of UserB FindMe File 101 with the name of UserB FindMe File 101 as listed in Trusted Member List 140
Trusted Member List 140
UserB FindMe File 101 IP xxx xxx xxx.xxx
UserC FindMe File 102 IP unknown
UserD FindMe File 103 IP unknown
call "CPDecrypt"
decrypt UserB FindMe File 101 using UserB Public Key 81 and associate UserB IP address with the name of UserB FindMe File 101 as listed in the Trusted Member List 140
Communications Means 30
Device "B" 11
User Interface 20
launch Peer-to-Peer Network Program 40
Operating System 25
Peer-to-Peer Network Program 40
run program
File Sharing Protocols 153
transmit to UserA the UserB FindMe File 101
FTP Protocol 150
HTTP Protocol 151
Storage 23
FindMe Folder 60
UserB FindMe File 101

Fig. 8

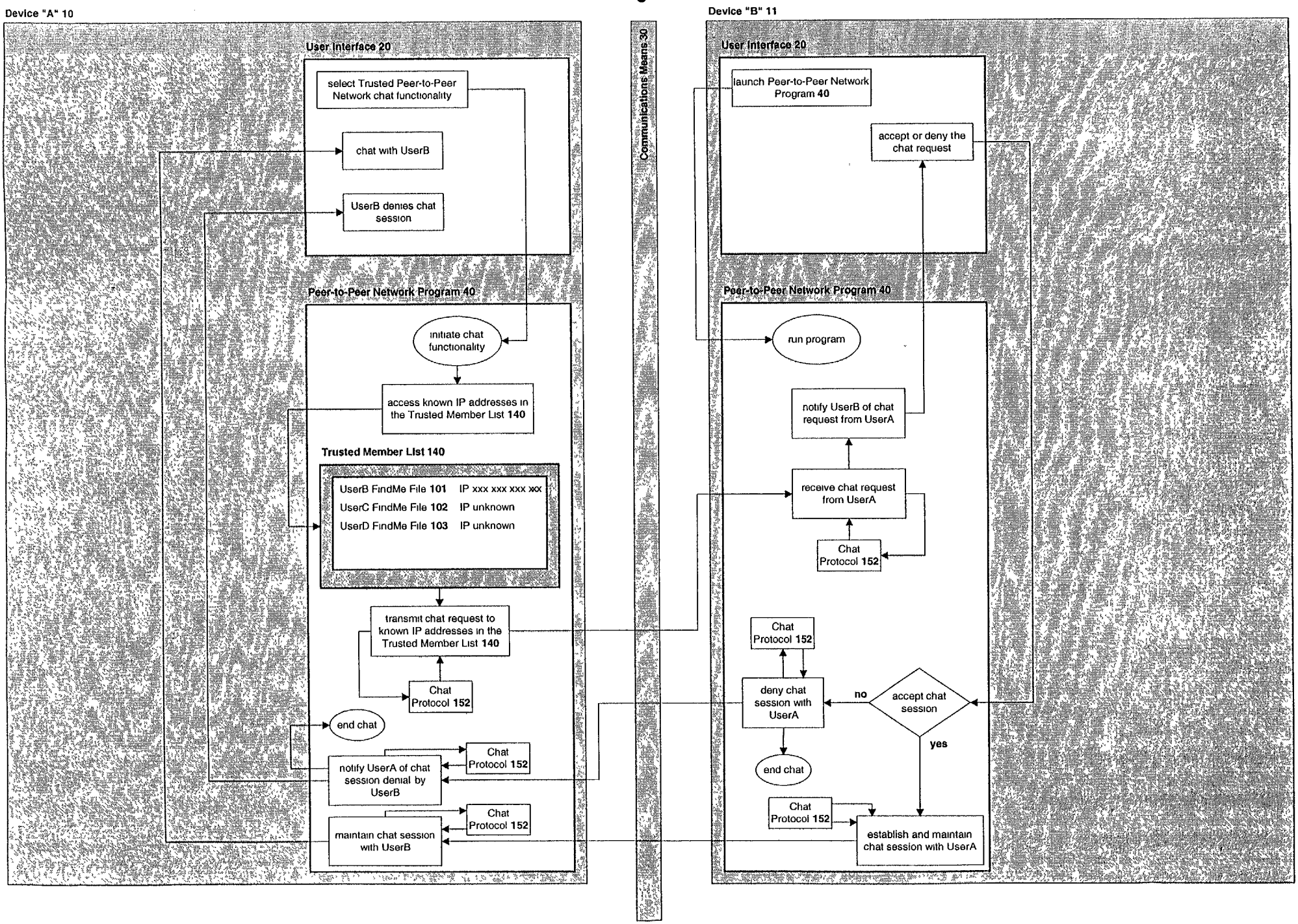
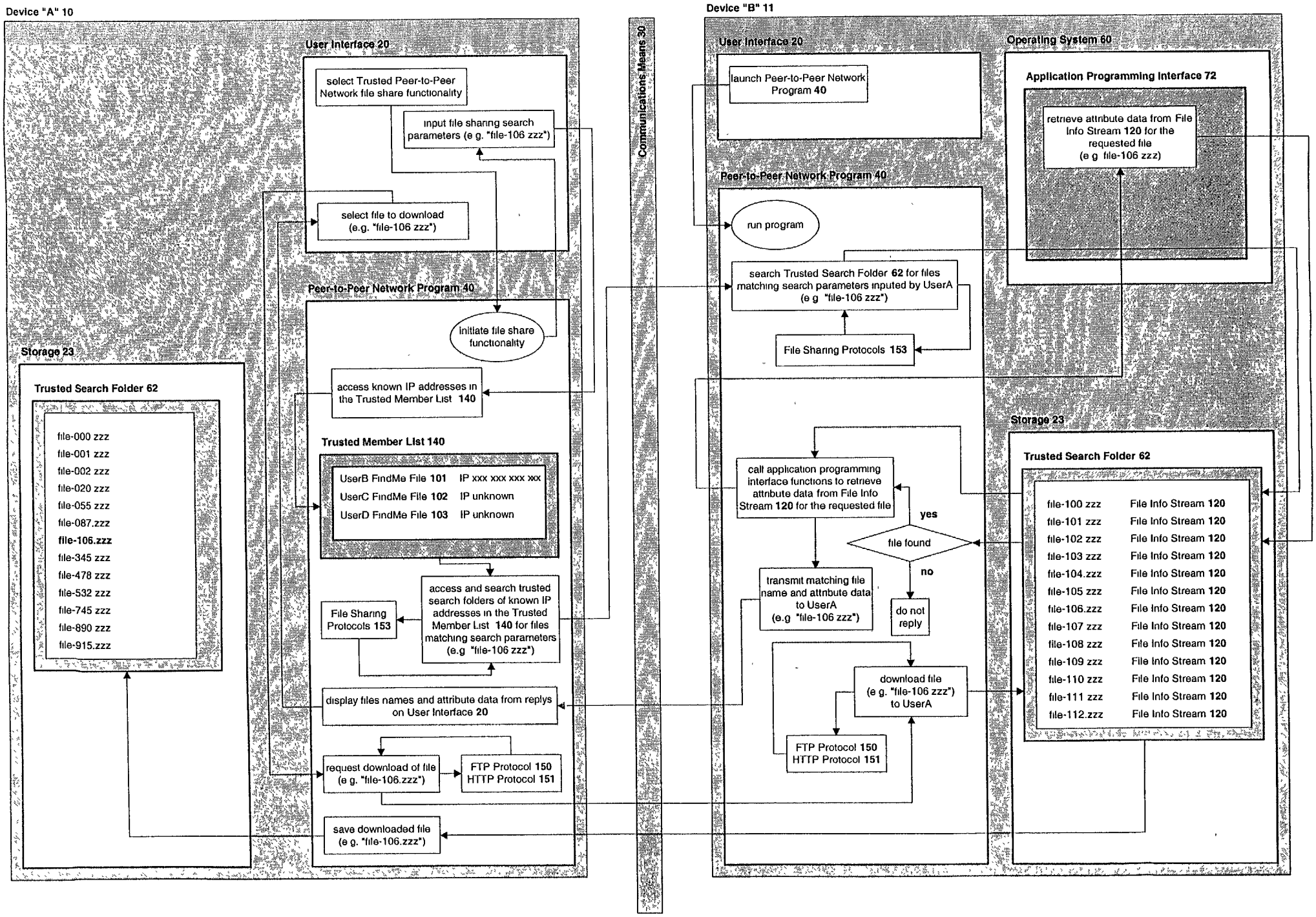

Device "A" 10
User Interface 20
select Trusted Peer-to-Peer Network chat functionality
chat with UserB
UserB denies chat session
Peer-to-Peer Network Program 40
initiate chat functionality
access known IP addresses in the Trusted Member List 140
Trusted Member List 140
UserB FindMe File 101 IP xxx xxx xxx xxx
UserC FindMe File 102 IP unknown
UserD FindMe File 103 IP unknown
transmit chat request to known IP addresses in the Trusted Member List 140
Chat Protocol 152
end chat
notify UserA of chat session denial by UserB
Chat Protocol 152
maintain chat session with UserB
Chat Protocol 152
Communications Means 30
Device "B" 11
User Interface 20
launch Peer-to-Peer Network Program 40
accept or deny the chat request
Peer-to-Peer Network Program 40
run program
notify UserB of chat request from UserA
receive chat request from UserA
Chat Protocol 152
Chat Protocol 152
deny chat session with UserA
no
accept chat session
yes
end chat
Chat Protocol 152
establish and maintain chat session with UserA

Fig. 9


Device "A" 10
User Interface 20
select Trusted Peer-to-Peer Network file share functionality
input file sharing search parameters (e g. "file-106 zzz")
select file to download (e.g. "file-106 zzz")
Peer-to-Peer Network Program 40
initiate file share functionality
access known IP addresses in the Trusted Member List 140
Trusted Member List 140
UserB FindMe File 101 IP xxx xxx xxx xxx
UserC FindMe File 102 IP unknown
UserD FindMe File 103 IP unknown
access and search trusted search folders of known IP addresses in the Trusted Member List 140 for files matching search parameters (e.g "file-106 zzz")
File Sharing Protocols 153
display files names and attribute data from replys on User Interface 20
request download of file (e g. "file-106.zzz")
FTP Protocol 150 HTTP Protocol 151
save downloaded file (e g. "file-106.zzz")
Storage 23
Trusted Search Folder 62
file-000 zzz
file-001 zzz
file-002 zzz
file-020 zzz
file-055 zzz
file-087.zzz
file-106.zzz
file-345 zzz
file-478 zzz
file-532 zzz
file-745 zzz
file-890 zzz
file-915.zzz
Communications Means 30
Device "B" 11
User Interface 20
launch Peer-to-Peer Network Program 40
Peer-to-Peer Network Program 40
run program
search Trusted Search Folder 62 for files matching search parameters inputed by UserA (e g "file-106 zzz")
File Sharing Protocols 153
call application programming interface functions to retrieve attribute data from File Info Stream 120 for the requested file
file found
yes
no
transmit matching file name and attribute data to UserA (e.g "file-106 zzz")
do not reply
download file (e g. "file-106 zzz") to UserA
FTP Protocol 150 HTTP Protocol 151
Operating System 60
Application Programming Interface 72
retrieve attribute data from File Info Stream 120 for the requested file (e g file-106 zzz)
Storage 23
Trusted Search Folder 62
file-100 zzz File Info Stream 120
file-101 zzz File Info Stream 120
file-102 zzz File Info Stream 120
file-103 zzz File Info Stream 120
file-104.zzz File Info Stream 120
file-105 zzz File Info Stream 120
file-106.zzz File Info Stream 120
file-107 zzz File Info Stream 120
file-108 zzz File Info Stream 120
file-109 zzz File Info Stream 120
file-110 zzz File Info Stream 120
file-111 zzz File Info Stream 120
file-112.zzz File Info Stream 120

United States Patent & Trademark Office

Office of Initial Patent Examination -- Scanning Division

SCANNED, #20

Application deficiencies were found during scanning:

☐ Page(s)_____ of _____ were not present for scanning. (Document title)

☐ Page(s)_____ of _____ were not present for scanning. (Document title)

☒ Scanned copy is best available.

DRAWINGS